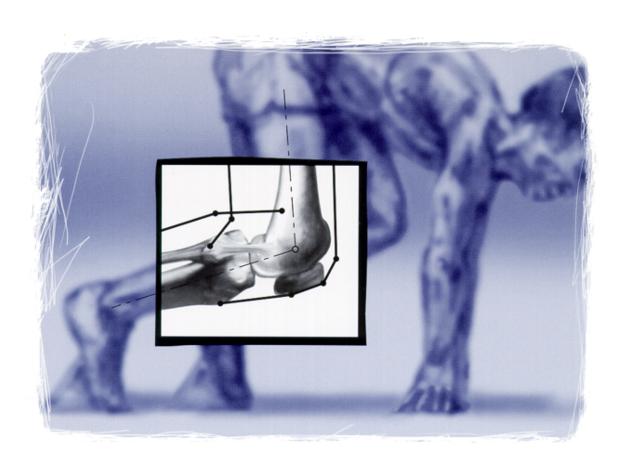


User Guide





SIMM 7.0

for Windows®

User Guide

September 2013

Part # SPC-70001

Important Notice

The information in this manual is subject to change without notice and should not be construed as a commitment by MusculoGraphics, Inc.

The software described in this manual is furnished under a license agreement and may be used or copied only in accordance with the terms of such agreement.

Copyright © 1992-2013 Motion Analysis Corp.

All rights reserved. This manual may not be reproduced in whole or in part without the expressed permission of Motion Analysis Corp.

MusculoGraphics

a division of Motion Analysis Corp.

3617 Westwind Blvd. Santa Rosa, CA 95403 Phone: +1 (707) 579-6500 Fax: +1 (707) 578-8473

E-mail: support@musculographics.com http://www.musculographics.com

Trademarks

Adobe Postscript is a registered trademark of Adobe Systems Incorporated. Windows is a registered trademark of Microsoft Corp. QuickTime is a registered trademark of Apple Computer, Inc.

Credits

Program Creation & Documentation:

Peter Loan Krystyne Blaikie Scott Delp Kenny Smith

Cover Illustration:

Idd Delp

Technical Support

+1 (707) 579-6500 x3 support@musculographics.com

Acknowledgements

Development of the initial version of SIMM was funded by the Rehabilitation Research & Development Center, Veterans' Affairs Medical Center, Palo Alto, CA.

We would like to thank Frans C.T. van der Helm, Ph.D., of the Delft University of Technology, for his contribution of muscle-wrapping algorithms.

We also thank Stephen J. Piazza, Ph.D., and Ryan Landon, M.S., of Pennsylvania State University, for their contribution of the NURBS-based contact detection algorithm.

Table Of Contents

Chapter 1: Introduction	1
1.1: SIMM - What it is	1
1.2: Applications	2
1.3: Design Goals	
1.4: The SIMM Environment	
1.5: About the Manual	
1.6: SIMM Tutorials	
Chapter 2: Tools	7
2.1: Introduction	
2.1.1: Definitions	
2.2: Opening Files	11
2.2.1: Drag and Drop	15
2.3: Saving Files	17
2.4: Model Viewer	20
2.4.1: Selector Menu	
2.4.2: Command Menu	22
2.4.3: Forms	28
2.4.4: Sliders	30
2.4.5: Toggle Buttons	31
2.4.6: Model-Window Viewing Commands	31
2.5: Plot Maker	37
2.5.1: Selector Menu	37
2.5.2: Command Menu	39
2.5.3: Generating Curves from a Model	40
2.5.4: Plotting Data from Motion Files	41

2.5.5: Y-variables	42
2.5.6: X-variables	46
2.5.7: Options Form	47
2.5.8: Toggle Buttons	50
2.6: Joint Editor	53
2.6.1: Selector Menu	53
2.6.2: Command Menu	53
2.6.3: Dof Form	55
2.6.4: Transform Order	56
2.6.5: Rotation Axes Form	56
2.7: Muscle Editor	57
2.7.1: Selector Menu	57
2.7.2: Command Menu	58
2.7.3: Global Parameter Commands	60
2.7.4: Force Parameters Panel	61
2.7.5: Attachments Panel	64
2.7.6: Current Path Panel	
2.7.7: Selected Attachments Panel	67
2.7.8: Model-Window Commands	68
2.8: Gencoord Editor	71
2.8.1: Selector Menu	72
2.8.2: Command Menu	72
2.8.3: Toggle Buttons	73
2.8.4: Gencoord Form	74
2.8.5: Inverse Kinematics	76
2.9: Plot Viewer	78
2.9.1: Pop-up Menu	78
2.9.2: Plot-Window Commands	80
2.9.3: Live Plot Curves	81
2.10: Wrap Editor	82
2.10.1: Selector Menu	83

2.10.2: Command Menu	83
2.10.3: Wrap Object Attributes	86
2.10.4: Transforming Wrap Objects	88
2.10.5: Model-Window Commands	89
2.10.6: Ellipsoid Wrapping Methods	90
2.10.7: Wrapping Tips & Techniques	92
2.11: Deform Editor	93
2.11.1: Selector Menu	95
2.11.2: Command Menu	96
2.11.3: Deform Object Attributes	97
2.11.4: Model-Window Commands	99
2.11.5: Transforming Deform Objects	101
2.11.6: Combining Multiple Deform Objects	103
2.11.7: Deform Tips, Techniques, and Caveats	107
2.12: Marker Editor	108
2.12.1: Selector Menu	110
2.12.2: Command Menu	110
2.12.3: Marker Parameters	111
2.12.4: Global Parameters	113
2.12.5: Model-Window Commands	113
2.13: Constraint Editor	116
2.13.1: Selector Menu	116
2.13.2: Objects Command Menu	117
2.13.3: Constraint Object Attributes	120
2.13.4: Model-Window Commands	121
2.13.5: Transforming Constraint Objects	123
2.13.6: Points Command Menu	124
2.13.7: Constraint Point Attributes	125
2.14: Segment Editor	126
2.14.1: Selector Menu	126
2.14.2: Command Menu	128

2.14.3: Segment Parameters	129
2.14.4: Model-Window Commands	134
2.15: Bone Editor	137
2.15.1: Selector Menu	138
2.15.2: Command Menu	138
2.15.3: Boolean and Read/Write Options	142
2.15.4: Norm Options	142
2.15.5: Model-Window Commands	145
2.16: Dynamics Tool	149
2.16.1: Selector Menu	153
2.16.2: Input and Output Options	153
2.16.3: Simulation Time	154
2.16.4: Motion for Inverse Dynamics	155
2.16.5: Smoothing (Low-Pass Cutoff)	156
2.16.6: System Parameters	156
2.16.7: Simulation Execution	158
2.16.8: Output Variables	159
2.17: Motion Editor	163
2.17.1: Selector Menu	163
2.17.2: Command Menu	163
2.17.3: Cropping a Motion	
2.17.4: Motion Events	166
2.17.5: Playback Options	167
2.18: Motion Reporter	172
2.18.1: Generating a Report	174
2.18.2: Selector Menu	176
2.18.3: Command Menu	176
2.18.4: Forms	179
2.18.5: Toggle Buttons	180
2.19: Skin Editor	181
2.19.1: Selector Menu	182

2.19.2: Command Menu	183
2.19.3: Skin Attachments	185
2.19.4: Group Weighting	186
2.19.5: Model-Window Commands	187
2.20: Function Editor	189
2.20.1: Command Menu	191
2.20.2: Function Editing	192
2.21: Scripting Tool	195
2.21.1: Top-Level Commands	197
2.21.2: Model Viewer Commands	200
2.21.3: Plot Maker Commands	203
2.21.4: Motion Reporter Commands	206
2.21.5: Dynamics Tool Commands	208
2.21.6: Motion Editor Commands	215
2.21.7: Muscle Editor Commands	218
Chapter 3: Input Files	221
Chapter 3: Input Files	
	221
3.1: Introduction	221
3.1: Introduction	221 222 224
3.1: Introduction	221 222 224 225
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII	221 222 224 225 227
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII 3.2.3: SIMM Binary	
3.1: Introduction 3.2: Bone Files	
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII 3.2.3: SIMM Binary 3.2.4: Wavefront OBJ 3.2.5: STL	
3.1: Introduction 3.2: Bone Files	
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII 3.2.3: SIMM Binary 3.2.4: Wavefront OBJ 3.2.5: STL 3.2.6: Collada 3.2.7: IGES	
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII 3.2.3: SIMM Binary 3.2.4: Wavefront OBJ 3.2.5: STL 3.2.6: Collada 3.2.7: IGES 3.3: Joint Files	
3.1: Introduction 3.2: Bone Files 3.2.1: SIMM ASCII 3.2.2: "Old" SIMM ASCII 3.2.3: SIMM Binary 3.2.4: Wavefront OBJ 3.2.5: STL 3.2.6: Collada 3.2.7: IGES 3.3: Joint Files 3.3.1: Body Segments	

3.3.5: Functions	249
3.3.6: World Objects	254
3.3.7: Materials	256
3.3.8: Cameras	257
3.3.9: Wrap Objects	259
3.3.10: Deform Objects	263
3.3.11: Deformities	267
3.3.12: Constraint Objects	270
3.3.13: Model Poses	275
3.4: Muscle Files	277
3.4.1: Muscles	277
3.4.2: Muscle Surfaces	284
3.4.3: Ligaments	286
3.5: Motion Files	286
3.5.1: Motion File Header	289
3.5.2: Gencoords and Muscles	293
3.5.3: Motion Objects	293
3.5.4: Other Data	296
3.5.5: Standard Deviation Data	297
3.5.6: Auxiliary Data	297
3.6: Plot Files	298
3.7: Model Archive Files	300
3.8: Script Files	301
3.9: Movie Files	303
Chapter 4: Norm	305
Chapter 5: Motion Module	311
5.1: Introduction	311
5.2: Opening Tracked Marker Files	312
5.2.1: Selecting Tracked Marker Files	

5.2.2: Tracked Marker Options Dialog	314
5.2.3: Analog Data	321
5.2.4: Real-time Import	324
5.2.5: IKS Files	327
5.3: Using the Mocap Model	328
5.3.1: The Mocap Model	329
5.3.2: The Static Pose	331
5.3.3: Calculation of Joint Centers	335
5.3.4: Scaling the Mocap Model	336
5.3.5: The Marker Set	338
5.4: Analog Configuration Files	340
5.4.1: importVariables.txt	341
5.4.2: forcepla.cal	345
Chapter 6: Tips on Using SIMM	A-1
Chapter 7: SIMM Resources	В-1
7.1: Preferences	B-1
7.2: Colors	B-16
7.3: Password.txt	B-16
7.4: SystemInfo.txt	B-17
7 5. Rongs	P 17

Index

1 Introduction

1.1 SIMM - What it is

SIMM (Software for Interactive Musculoskeletal Modeling) is a software system that enables you to create and analyze graphics-based models of the musculoskeletal system. In SIMM, a musculoskeletal model consists of a set of bones that are connected by joints. Muscle-tendon actuators and ligaments span the joints. The muscles and ligaments develop force, thus generating moments about the joints.

SIMM allows you to analyze and test a musculoskeletal model by calculating the moment arms and lengths of the muscles and ligaments. Given muscle activations, the forces and joint moments (muscle force multiplied by moment arm) that each muscle generates can be computed for any body position. By manipulating a model on the computer graphics system, you can quickly explore the effects of changing musculoskeletal geometry and other model parameters.

Since the software can be used to study many different musculoskeletal structures, it can enhance the productivity of investigators working on diverse problems in biomechanics. SIMM provides a framework that organizes the parameters of a model and allows people to work together on a modeling project. The moving, three-dimensional images of anatomical structures that you can create are extremely valuable when developing a model and when communicating the results of an analysis.

1.2 Applications

SIMM has a wide variety of applications. Here are a few examples.

Biomechanics researchers are using SIMM to create models of the human elbow, wrist, jaw, and other anatomical structures. These models can be altered according to particular surgical procedures to study how the surgical alterations affect muscle function. SIMM can also be used to analyze and display the mechanics of injuries.

Neuroscientists are using SIMM to study how the central nervous system controls movement. For example, muscle activation patterns determined from electromyographic recordings can be used to estimate muscle forces and joint moments generated during a task. The computed joint moments can then be compared to experimentally recorded moments.

Medical students and residents use models created with SIMM to study musculoskeletal anatomy and function. In addition to visualizing anatomical structures, students gain an appreciation for the interplay of muscle architecture and joint geometry.

Kinesiologists who record and analyze the motion of persons with movement disabilities are using SIMM to create three-dimensional animations of a person's movement. Movements, such as walking, can be quantitatively compared to normal movement to gain insight into the causes of movement deformities. Motion can also be analyzed in the context of optimizing athletic performance.

Human factors engineers who need to account for muscle strengths when designing products or work stations can use

SIMM to study how posture effects muscle strength. Limits on joint ranges of motion can also be taken into account.

Biologists interested in animal movement can create models to quantify limb function. Investigating movement strategies in other species can provide insights needed to design machines that move.

Computer scientists who develop models of the human body for virtual environments can use SIMM to create the models and compare them with biomechanical data for verification

Animators can use SIMM to develop realistic representations of human and animal movements. World objects can be added to provide a context for the animation.

1.3 Design Goals

We established four goals in designing and implementing the software. Specifically, it should:

- (i) be *general* enough so that a wide variety of musculoskeletal structures can be modeled,
- (ii) provide *realistic* models of muscle and tendon, and allow accurate specification of joint kinematics,
- (iii) provide an *interactive*, graphics-based environment so the model can be visualized, altered, analyzed, and tested efficiently, and
- (iv) be *extensible* so that new features, such as a more complex muscle-tendon model, can easily be added to the software.

1.4 The SIMM Environment

SIMM allows you to load one or more musculoskeletal models by reading sets of input files. Once loaded, a model can be acted upon by a number of editing and analysis tools (Figure 1-1). Each tool is contained within its own window and has a distinct function, such as controlling the viewing of the models (Model Viewer), altering the muscles (Muscle Editor), and making plots for analysis (Plot Maker).

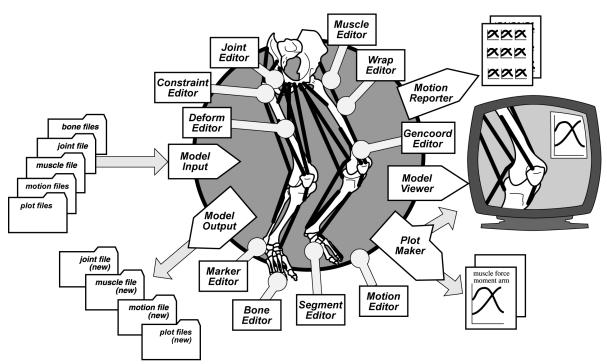


Figure 1-1. Structure of SIMM. Input files describing the bone surfaces (bone files), joint kinematics (joint file), and muscle-tendon parameters (muscle file) are read in to make a musculoskeletal model (motion files can also be read in to animate the model). A model can be altered using the Joint Editor, Muscle Editor, Gencoord Editor, Wrap Editor, Deform Editor, Model Viewer, Skin Editor, and Function Editor. Information is extracted from the model by making plots, movies, or by exporting edited joint, muscle, and motion files. The Motion Reporter creates reports for gait analysis and study of other motions. The Dynamics Tool and Motion Module (not shown) are optional modules that allow you to perform dynamic simulations and import motion capture data, respectively.

A musculoskeletal model is specified with three types of input files. The *bone* files contain lists of the polygons representing the bone surfaces. The *joint* file specifies the kinematics of each joint. Finally, the *muscle* file contains a list of coordinates that describe the line of action of each muscle-tendon actuator, and the parameters needed to compute muscle force. Ligaments can also be defined in the muscle file. SIMM scans these input files and creates a data structure that represents the musculoskeletal model. There are also *motion* files that contain joint angle and other data representing a motion, and *plot* files, which contain any data that can be calculated by the Plot Maker. Each of these types of files is described in detail in Chapter 3, Input Files.

1.5 About the Manual

This manual describes how to use SIMM to develop and analyze musculoskeletal models. Chapter 2, Tools, describes the user interface and the 15 software tools. Chapter 3, Input Files, details how to define a musculoskeletal model. Chapter 4, Norm, describes the utility program that you use to preprocess bone files before loading them into SIMM. Chapter 5, Motion Module, describes an optional module that allows you to easily import data recorded by a motion capture system. Appendix A contains some useful tips and caveats on using SIMM. Appendix B describes how to customize SIMM to better fit your needs and preferences. We recommended that you read both of the appendices before using SIMM to make your own musculoskeletal models.

1.6 SIMM Tutorials

SIMM comes with sample leg, arm, and neck models built into the software, and includes several tutorials that will introduce you to the basic capabilities of SIMM. The tutorials are located in the Help menu, accessible from the SIMM menu bar. There are four tutorials: Viewing a Musculoskeletal Model, Plotting Muscle Properties, Joint Editing, and Gait Analysis. There are also demos of the Motion Module and the Dynamics Pipeline.

A more complete version of the demo leg model is provided with the full version of SIMM. You can use this model for a musculoskeletal analysis of the lower extremity, or as a template for building your own model. For more details on the development and validation of the leg model, consult the following publications:

Delp, S. L., Loan, J. P., "A Computational Framework for Simulating and Analyzing Human and Animal Movement," *IEEE Computing in Science and Engineering*, vol. 2, 2000, pp. 46-55.

Delp, S. L., Loan, J. P., "A Graphics-Based Software System to Develop and Analyze Models of Musculosk-eletal Structures," *Computers in Biology and Medicine*, vol. 25, No. 1, 1995, pp. 21-34.

Delp, S. L., "Surgery Simulation: A Computer Graphics System to Design and Analyze Musculoskeletal Reconstructions of the Lower Limb," Ph.D. Dissertation, Stanford University, 1990.

PDF versions of these publications, as well as other articles describing research done with SIMM, can be found here:

http://www.stanford.edu/group/nmbl/publications/index.htm

2 Tools

2.1 Introduction

SIMM has 13 tools that enable you to create, modify, and analyze your musculoskeletal models. The tools are: Model Viewer, Plot Maker, Muscle Editor, Joint Editor, Gencoord Editor, Segment Editor, Bone Editor, Constraint Editor, Wrap Editor, Deform Editor, Motion Editor, Marker Editor, Function Editor, Skin Editor, and Motion Reporter. Also, if you have the Dynamics Pipeline Module or FIT Module options to SIMM, there is a Dynamics Tool for interfacing SIMM to dynamic simulations. There is also a Scripting Tool that enables you to automate some of the functions in SIMM. This chapter gives a detailed description of each tool.

The tools use *menus*, *forms*, *sliders*, and *toggle buttons* to give you access to a variety of information. Menus are used to make selections (*e.g.*, choosing a muscle). Forms are used to enter information (*e.g.*, entering a title). Sliders let you change the value of something (*e.g.*, a generalized coordinate) smoothly. Toggle buttons are used to control the state (yes/no or on/off) of a parameter.

2.1.1 Definitions

menu

A *menu* is a vertical list of options or commands, each enclosed in a box. You can select a menu option by clicking the left mouse button in its box. The command is per-

formed as soon as you press the mouse button, not when you release it. Some menu items display pop-up submenus when they are selected. Menu items that do this can be differentiated from other menu items by an angle bracket (greater-than sign) on the right side of the item box. For example, the model selector, which displays a pop-up menu of the models, is labeled **model** >.

muscle menu

A muscle menu is a special type of menu. Muscle menus group muscles according to group names assigned in the muscle file (e.g., hip flexors, see Section 3.4, Muscle Files). When you click the left mouse button on one of the items in a muscle menu, the corresponding muscle is selected and the box to the left of the muscle name is highlighted. When you click the left mouse button on that item again, the muscle is unselected. Several of the tools display the muscle menus, but selecting a muscle in one tool's menu does not mean that it is selected in another tool's menu. For example, the Model Viewer and Plot Maker each display separate copies of the same muscle menus. When you select a muscle in the Model Viewer, it changes the model view, but does not select the muscle within the Plot Maker. If you want to generate a plot curve for that muscle, you must also select it from the muscle menu in the Plot Maker.

To make it easier to select and unselect large groups of muscles, the title bar of each muscle menu also has a special function. If one or more muscles in a menu are unselected, then clicking the left mouse button in the title bar of that menu causes all of the muscles to be selected. If all of the muscles are already selected, then when you click in the title bar they become unselected. The last item in each muscle menu is labeled **<<done>>**. Selecting **<<done>>** with the left mouse button "puts away" the menu so that it is no longer displayed. If any of the muscles in the menu are selected when you put away the menu, they remain selected even

though you can no longer see the menu. When you are finished selecting muscles from a particular muscle menu, you do not have to put the menu away by selecting <<done>>>. Doing so is merely a convenience so that you can display other muscle menus without taking up too much space in the tool window.

form

A form consists of a group of fields that can be edited. A field is a boxed region that contains a number or other text. You can position the cursor anywhere in a field by clicking the left mouse button at the desired position. A text cursor appears at this location to indicate that any text you type in will be inserted at this location. You can use the left and right arrow keys to move the text cursor one character at a time. You can highlight a section of the text field that you want to delete or change by "sweeping" the cursor over the region. To do this, press the left mouse button at one end of the region, move the cursor to the other end, then release the left mouse button. You can adjust the size of this highlighted region by holding down the shift key while pressing the left and right arrow keys to move one character at a time. If some text is highlighted when you type in a character, the highlighted text is deleted and the typed character is put in its place. To delete the character just before the text cursor, press the Backspace or Delete key. You can copy the text from a text field to the Windows clipboard by selecting it and pressing Ctrl-C. You can also paste text from the clipboard into a text field by pressing Ctrl-V.

Some form fields can contain only numbers (*e.g.*, the gencoord form in the Model Viewer). When editing a numerical field, you can enter only numbers, a decimal point, and a minus sign. You cannot enter exponents.

When you are done editing the text in a field, press the Enter key, or press the left mouse button while the cursor is outside the field (and not over any other menu item or form

field). If the field is a numerical one, SIMM will check to make sure that the number you have entered is valid. If it is not, SIMM will restore the previous value. If the field is empty when you press return, SIMM will restore the text that was in the field before you began editing it. You can also press the Tab key when you are done editing a field. This will move the text cursor to the next field in the form and select all of the text within it.

slider

Sliders in SIMM are just like sliders or scroll bars in other programs. Each has a thumb which you can click on with the left mouse button and drag along the shaft of the slider. You can also click the left mouse button on either of the two arrow buttons to change the value more slowly. If you click in the slider shaft away from the thumb, the thumb will jump to where you clicked.

toggle button

Toggle buttons control the on/off (yes/no) state of various parameters. Clicking the left mouse button in the box toggles the state between the two values. When the state is on (yes), the toggle button is yellow. When the state is off (no), the button is gray.

selector menu

All tools have a selector menu at the top of the tool window. The text above this menu shows the current model (the model that the tool is currently working on), and the current plot (the plot that the tool is working on). Below this text are "selectors" for changing the current model or plot. You can change the current model with the model selector, which is the box labeled **model**. Clicking the left mouse button in this box pops up a menu of the models you have created. You can then choose the one you want by releasing the button while the cursor is over its name. You only need to use the model selector if you have more than one model at a time loaded into SIMM. You can select the current plot in a similar fashion using the plot selector (labeled **plot**). If a tool does not operate on specific models (e.g., Plot Viewer),

the model selector is not displayed in the header region. Likewise, if the tool does not operate on specific plots (e.g., Model Viewer), the plot selector is not displayed. Also in this region of every tool is a help button at the far right. Click the left mouse button on it to open a window with some helpful text.

In general, the interface for each of the tools is confined to the tool window. That is, the tool window must be the active window for you to perform any of that tool's commands. However, three of the tools (Model Viewer, Muscle Editor, and Plot Viewer) can recognize key and mouse button presses when they occur in the model or plot windows. For example, to select muscle points in the Muscle Editor, you press the Space Bar and click the left mouse button *in the model window*. This feature allows you to operate on the models and plots without having to open the tool window. In some cases the same command can be performed in the tool window as well, in other cases the command can only be performed in the model or plot window. See the relevant tool section in this chapter for more details.

2.2 Opening Files

To load files into SIMM, you can use either the File menu in the menu bar, or "drag and drop" them from a Windows File Explorer window to the SIMM window. To use the File menu, select *File > Open...* and use the Windows file browser to locate the files. By default the file browser displays all files with SIMM file name extensions. You can use the *Files of type* popup menu at the bottom of the browser to limit the display to files of a single type to make it easier to locate a specific file. It is possible to select and open multiple files at once using the file browser. This can be useful

when working with multiple models, motions or plots, or when applying different muscle files to a single model.

joint and muscle files

To define a musculoskeletal model, all of the model components except the muscles (*e.g.*, segments, joints, gencoords, wrap objects, kinematic functions, materials) must be defined in a joint file. If muscles are to be part of the model, they must be defined in a separate muscle file (see Chapter 3, Input Files, for a description of the format of each type of file).

If you select a single joint file to open, SIMM will create a new model based on that file. If a muscle file name is specified within the joint file (using the *muscle_file* parameter), that muscle file will be loaded as well. If no muscle file name is specified, the model will be loaded without any muscles, unless a muscle file is also selected in the file browser. If a muscle file is selected in addition to a joint file in the browser, SIMM will load the model from the joint file and add to it the muscles from the muscle file (thus overriding the *muscle_file* parameter, if specified).

If you select multiple joint files in the file browser, SIMM will open each one and create a new model from it. However, when selecting more than one joint file you cannot select more than one muscle file as well because SIMM has no way to determine which muscle file corresponds to which joint file. If you want to load multiple joint files at once, you should specify the corresponding muscle file names within each joint file.

bone files

A bone file defines a mesh of polygons representing the surface of an object. Bone files in SIMM are used to define not only anatomical bones, but also world objects, motion objects, force mattes, spring floors, and contact objects. You cannot open bone files using the File menu in SIMM (though you can load them into a model using the Bone Edi-

tor; see Section 2.15 for details). Instead, bones are referenced from within a joint file and loaded automatically when the joint file is loaded. To locate bone files for bones referenced in a joint file, SIMM searches the folders in the following list, in the order shown. If a particular folder does not exist or the bone file is not found in it, SIMM proceeds to the next folder in the list.

- (1) the folder specified by the bone_path parameter in the joint file
- (2) the folder *bones* in the folder containing the joint file
- (3) the folder containing the joint file
- (4) the folder specified by the BONES_FOLDER preference (this preference, if not defined by the user, defaults to *Simm\Resources\bones*)
- (5) the folder *Simm*\Resources\mocap\bones

motion files

When you select a motion file in the file browser, SIMM will link it to the current model. If a model window is the topmost window, that model is considered the current model. If a tool window is the topmost window, whatever model that tool is set to is considered the current model. When the motion file is loaded, SIMM matches the column names in the motion to the names of the gencoords, muscles, markers, and bodies in the model. Any column that does not match one of these elements is labeled as "other data." Other data can be plotted in the Plot Maker, but is not used to control the model when the motion is animated. Once a motion has been linked to a model, it behaves much like a true generalized coordinate. That is, you can animate the model display by pressing the keys specified in the motion file, or moving the motion slider in the Model Viewer (see Section 2.4, Model Viewer). You can also use the motion variable as the independent variable when making plots (see Section 2.5, Plot Maker).

plot files

When you select a plot file in the file browser, the way that SIMM processes it depends on which SIMM window was topmost when you selected *File* > *Open...* If a plot window was topmost, SIMM will read the plot file and add the curves to the selected plot. If some other window was topmost, SIMM will create a new plot window and add the curves to it. In this way multiple plots from separate files can be combined into a single plot if desired.

model archive files

A model archive is a compressed file containing all of the files associated with a model, including joint file, muscle file, bone files, and even motion files that are linked to the model. If you select multiple model archives in the file browser, SIMM will open each one and create a new model from it.

script files

A script file is a text file containing SIMM scripting commands. These commands (described in Section 2.19, Scripting Tool) can be used to load and save models, process motion data, and specify settings in the tools. You can create your own script file with a text editor by typing commands into a file. SIMM can also automatically generate script files, as described in Section 3.8, Script Files. Script files that SIMM creates have the .ssc suffix. If you make your own script file you can use the .ssc or .txt suffix When you load a script file using the File menu, SIMM executes the commands in it using the Scripting Tool. Any commands in the file that are model-specific (e.g., specifying settings in the tools) are executed with respect to the model that was current when you selected File > Open... If there is no current model, model-specific commands in the script file are ignored.

2.2.1 Drag and Drop

All of the types of input files described in the previous section, plus those used by the Motion Module, can be loaded into SIMM by dragging them from a Windows file explorer window and dropping them onto a SIMM window or the SIMM background. Depending on the type of file, SIMM uses the specific location of where you dropped it to determine how the file should be imported. Here are the rules SIMM follows for drag and drop:

One or more JNT files can be dropped onto any SIMM window or the background. SIMM will load each one as a new model.

A single MSL file can be dropped onto SIMM only if dropped with a single corresponding JNT file. The JNT file will be loaded as a new model, using the dropped MSL file for the muscle definitions instead of the muscle file specified in the JNT file (if any).

SMM One or more model archive files can be dropped onto any SIMM window or the background. SIMM will load each one as a new model.

One or more motion capture files can be dropped onto SIMM. If the files are dropped onto the SIMM background window, each file will be treated as a static trial, and used to load a new SIMM model (using the Mocap Model as the generic model). If the files are dropped onto a SIMM model window (or tool window set to a specific model), they will be treated as motion trials and will be loaded onto that model. Note: the Motion Module or C3D Module is needed for this feature.

One or more plot files can be dropped at the same time. If they are dropped onto a plot or plotkey window, they will be

added to that plot. If they are dropped anywhere else a new plot will be created to hold the data.

One or more motion files can be dropped onto a SIMM model window or tool window set to a specific model. They will be added to the model associated with that window.

One or more bone files can be dropped onto a SIMM model window. They will be added as bones to the body segment on which they are dropped. If they are dropped onto something in the model window that is not associated with a specific body segment, they will be added to the ground segment.

A single script file can be dropped onto SIMM. If the file is dropped onto a model window (or any window if there is only one model loaded), the model-specific commands in the file are executed with respect to that model (*e.g.*, specifying settings in the tool windows). If there are multiple models loaded into SIMM and the script file is dropped onto a window not associated with a particular model (*e.g.*, background window, plot), SIMM will execute only the commands in the file that are not specific to a model.

A single image file can be dropped onto a SIMM model window in order to change a texture. The pixel on which it is dropped must be part of a bone or world object which already contains a texture. In this case the entire texture will be replaced with the image dropped onto it.

Movie files can be dropped one at a time onto a SIMM model window, to link them to a motion. If more than one motion is in the model, a dialog box will be displayed that allows you to select which motion to link the movie to. If no motions are in the model, an error message is printed.

Chapter 2: Tools

WMV AVI

MPG

MOV

Mp4

One or more IKS files can be dropped onto a SIMM model or a tool window set to a specific model. They will be treated as motion trials and will be loaded onto that model. Note: the Motion Module or C3D Module is needed for this feature

2.3 Saving Files

When saving joint, muscle, motion, and plot files to disk, you use the Windows file browser to specify the name and location of the file to write. Use the browser to navigate to the directory where you want the file to go, then either type the name of the file to create or select an existing file to be replaced.

When saving files to disk, SIMM does not preserve any comments that may have been present in the original input file. Be careful to avoid overwriting any existing SIMM files that contain useful comments.

joint and muscle files

To save a model's joints or muscles, either the model's window must be the active (*i.e.*, topmost) window, or if a tool window is active, it must be set to that model. To save the model's segments, joints, generalized coordinates, and kinematic functions, select *File* > *Save Joints...* To save the model's muscles, select *File* > *Save Muscles...* Use the Windows® file browser to specify the name and location of the joint or muscle file to be written. If you do not specify a *.jnt* or *.msl* file name extension for the file, SIMM will add the appropriate one to the file name you specify.

motion files

To create a motion file for a model, either the model's window must be the active (*i.e.* topmost) window, or if a tool window is active, it must be set to that model. Select *File* > *New Motion...* and use the Windows file browser to specify

simm simple series of the motion file to be created. Simm will automatically add a .mot extension to the file name for you if necessary. When you click "OK" in the file browser, the new motion file will be created with a motion header, but without any motion frames. To add motion frames to the new file you must repeatedly position the model by adjusting its generalized coordinate values, then select File > Add Motion Frame. Each time you select Add Motion Frame from the File menu, a new row of information containing the model's current generalized coordinates will be appended to the motion file.

When SIMM first creates a motion file, it gives the motion a default name and sets the datarows variable to 200 (see Section 3.5, Motion Files). After saving your motion to a file, you may want to change its name or the value of datarows. If the file contains fewer than 200 rows of data, SIMM will print a warning when it reads the file, but it will still load the motion correctly. Thus you can check the motion right after creating it by loading the file back into SIMM by selecting *File > Open...*

There is currently no way to edit frames of an existing motion and save the results to a file. The Motion Editor allows you to crop motions and save them to a file, but you cannot modify individual frames of data. To accomplish this task, you must use the Model Viewer to step through the motion frame by frame, altering the model configuration where desired, and writing each frame to a file with the *Add Motion Frame* command.

plot files

To save the contents of a plot window to a file, either the plot window must be the active (*i.e.*, topmost) window, or if a tool window is active it must be set to that plot. To save the plot, select $File > Save\ Plot...$ and use the Windows[®] file browser to specify the name and location of the plot file to be written. SIMM can write plot files in two formats: tab-

delimited columns of data, and Postscript. If the file name you specify has a .ps extension then SIMM will write the plot in Postscript format which can be sent directly to a Postscript printer. Otherwise SIMM will write the plot data in tab-delimited columns and add a .plt extension to the file name, if necessary. Tab-delimited plot files are designed to be easy to read or import into word processing or spreadsheet programs.

Note: SIMM cannot read plot files in Postscript format. Thus plots that you intend to load back into SIMM later should be saved in the tab-delimited .plt format.

model archive files

A model archive is a compressed file containing all of the files associated with a model, including joint file, muscle file, bones files, and any motion files currently linked to the model. To create a model archive, use the *File > Export Model Archive* command. The archive will contain everything needed to load the model back into SIMM at a later time. Because everything is in one file, it is a good way to share models with other users, or to backup your models before modifying bone files or other components that are typically shared among models.

bone files

Bone files can be edited and saved using the Bone Editor. This tool has a graphical interface to norm, as well as some simple vertex and polygon editing functions. After modifying a bone, you can save it to a file using the Bone Editor's write bone 1 command.

model script files

A model script file is a file of scripting commands that sets model-specific parameters in the SIMM tools. For example, it contains commands to set the output variables in the Motion Reporter and the plot creation parameters in the Plot Maker. If the *read_script_file* and *write_script_file* parameters are defined in the joint file (see Section 3.3 for details), SIMM will automatically read and write a script file for that

model. If you want to save a model script file manually (e.g., after specifying various tool settings for a model with read_script_file turned off), make the model window the active window and select File... Save Model Script from the menu bar. SIMM will create a script file in the same folder as the joint file, with the same base name as the model, but with the suffix ".ssc". The file will contain commands to restore the settings in the Model Viewer, Plot Maker, Motion Reporter, Motion Editor, and Dynamics Tool to the same values as when the file was created.

2.4 Model Viewer

The Model Viewer allows you to change the views of the musculoskeletal models. You can rotate, scale, and translate a model into any viewing perspective, as well as choose which muscles to display on the bones. There are also commands to display shadows of the body segments, and start or stop a continuous animation of a model (Figure 2-1). You can also create TIFF images or AVI movies of the model window using this tool. Some of the viewing commands that are available from within the Model Viewer window can also be performed by pressing keys while the model window is the active (*i.e.*, topmost) window. These key sequences serve as short cuts so that you can change the view of a model without having to open the Model Viewer window. These *model window viewing commands* are described in Section 2.4.6.

2.4.1 Selector Menu



The model selector lets you select which model you will change with the Model Viewer. The menus and forms in the Model Viewer window always correspond to the current

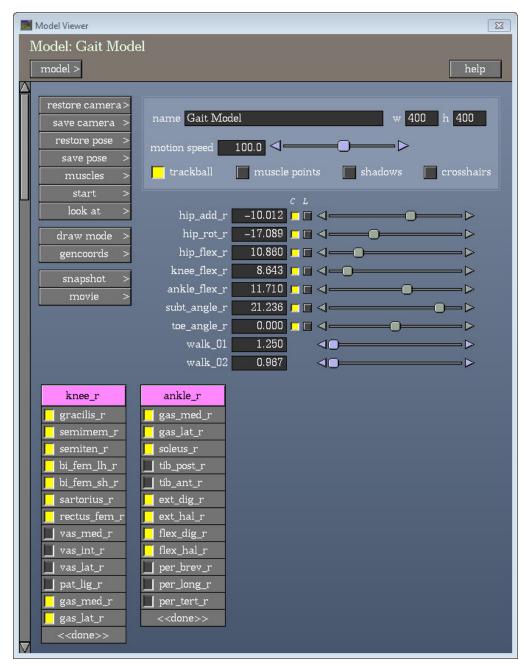


Figure 2-1. Model Viewer window

model (setting the current model does not affect the use of model window viewing commands). If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Model Viewer. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.4.2 Command Menu

restore camera

This command restores the camera view of the current model to one that you had previously saved. When you select this command a pop-up menu of the available views is displayed. Selecting one will restore the camera view to the one saved in that buffer. See the **save camera** command for directions on saving a camera view.

save camera

This command saves the current camera view of the model to a buffer. When you select this command a pop-up menu of the current view buffers is displayed. Selecting one will save the current model view to that buffer, overwriting the previous one. You can also select **make new camera** from the menu to create a new camera view. See the **restore camera** command for directions on how to restore the camera once you have saved it.

restore pose

This command restores the pose of the model to one that you had previously saved. A model pose contains the values and velocities of all of the gencoords, as well as the activations and display state (visible/invisible) of all of the muscles. When you select this command a pop-up menu of the available poses is displayed. Selecting one will restore the pose to the one saved in that buffer. See the **save pose** command for directions on saving a model pose.

save pose

>

>

This command saves the current pose of the model to a buffer. A model pose contains the values and velocities of all of the gencoords, as well as the activations and display state (visible/invisible) of all of the muscles. When you select this command a pop-up menu of the current poses is displayed. Selecting one will save the model pose to that buffer, overwriting the previous one. You can also select **make new pose** from the menu to create a new pose. See the **restore pose** command for directions on how to restore the pose once you have saved it.

muscles

This command lets you choose which muscles are displayed on the current model. When you select this item, a pop-up menu of the muscle groups is displayed. When you choose a muscle group from this menu, it will appear at the bottom of the Model Viewer window. If the chosen muscle group is already displayed at the bottom of the window, then selecting it will turn it off, so that it is no longer displayed. As you select muscles from the menus at the bottom of the tool window, the model window is redrawn to show the new muscle selections.

start

This command lets you automatically animate motions without having to press keys or mouse buttons. When you select it, a pop-up menu is displayed containing a list of the motions that have been linked to the current model. When you select one of the motions, the model begins at the current time value of that motion and proceeds at the rate specified in the *motion speed* field. The **start** button is changed to read **stop** to indicate that the motion is active. If the *loop motion* checkbox is turned off for this motion in the Motion Editor, the animation will stop when it reaches the last frame of motion data. If *loop motion* is on, the animation will continue until you click on the **stop** button. Because this procedure repeatedly draws the model in the model window, it can slow down the computer, making it difficult

to execute commands in other windows. So although you can start an animation and then open other tools and make plots, for example, this is not advised. You should also avoid changing the model view while the animation is proceeding. If you need to change the view of the model, stop the animation first, then restart it after positioning the model appropriately.

look at

This command lets you change the body segment that is fixed in space, for display purposes only. The *origin* of this segment will not move as you flex the joints in your model, but it may rotate about its origin as the joints move. This command is particularly useful if you want to zoom in on a body segment that is far from the origin of the ground frame, and then rotate the view "locally" around this body segment.

Note to Dynamics Pipeline users: The **look at** command does not affect the ground segment used for calculating dynamics. Regardless of which segment is treated as ground for display purposes, the dynamics ground segment will always be the fixed segment in the SIMM model (see Section 3.3.2, Fixed Segment, for more details).

drawmode

This command gives you control over the display of the body segments and world objects. When you select this command, a pop-up menu of the body segments is displayed. After selecting one of them (or one of the four groups of objects at the bottom of the menu), roll-off to the right to select one of drawing styles. The six styles are: gouraud shaded (filled polygons, smooth shading), flat shaded (filled polygons, flat shading), solid fill (filled polygons, no shading), outlined (white polygons with highlighted edges), wireframe (hollow polygons), and none (no display). Furthermore, if body segment groups are defined in the model, then the drawmode pop-up menu will be organized by segment groups. This makes it easy to quickly

change the draw mode of several body segments at a time. For example, if you are working with a multi-limbed model, you may want to group the body segments by limb. In doing so you could quickly change the drawmode of an entire limb without having to change the drawmode of each individual segment in the limb.

Note: There is another method for changing the drawmodes of bones and world objects. If you right-click on a bone or world object in the model window, SIMM will display a pop-up menu. Using this menu, you can change the object's drawmode and material, as well as turn on and off the display of the segment axes and object's normal vectors. You can also set the camera to "look at" that segment, as if you had used the **look at** command.

gencoords

This command allows you to control which gencoord sliders are visible in the Model Viewer window. If your model contains a large number of gencoords, it can be tedious to quickly locate an individual gencoord slider amongst all the others. When you select the **gencoord** command, a pop-up menu of the gencoords is displayed. Selecting items in this menu toggles the visibility of the gencoord's slider in the Model Viewer. Furthermore, if gencoord groups are defined in the model, then the gencoord menu will be organized by these groups. This makes it easy to quickly toggle the visibility of several gencoord sliders at a time.

Note: when a model is loaded, SIMM checks to see how many gencoords are in the model. If there are more than 50 gencoords in the model then SIMM hides all gencoord sliders by default to avoid filling the Model Viewer window with too much information.

snapshot >

The snapshot command allows you to easily capture still images from the SIMM model window in TIFF image for-

mat. When you select the **snapshot** command, a pop-up menu is displayed. The menu contains the following items:

choose snapshot file...

This menu item displays a file browser dialog that allows you to specify the name and location of the next snapshot image to be created by SIMM. By default SIMM will create snapshots starting with the filename <code>snapshot_0001.tif</code>. Each subsequent snapshot file will have the numeric portion of the filename incremented by one (e.g., <code>snapshot_0002.tif</code>, <code>snapshot_0003.tif</code>, etc.). When you select <code>choose snapshot filename...</code> to specify a different snapshot filename, SIMM examines the filename you enter to see if it contains a number. If so, SIMM uses that number as its snapshot counter. Finally, SIMM will make sure that the name of each snapshot file it creates ends with <code>.tif</code>. This identifies the file to other programs as a TIFF image file.

include alpha channel

This menu item allows you to specify whether SIMM should include an alpha channel in the snapshot images it creates. By default SIMM creates a 24-bit RGB image without an alpha channel. If this menu item is selected, SIMM will create a 32-bit RGBA image that includes z-depth values as an alpha mask. By including an alpha channel, SIMM snapshot images can be combined with other images of 3D scenes using third-party compositing software.

take snapshot F12

This menu item creates a single snapshot image of the current model window. If a file already exists with the current snapshot file name it is automatically overwritten (so be careful). The function key F12 will also take a snapshot of the active model window.

auto-snapshot

This menu item puts SIMM in *auto-snapshot* mode. In this mode, SIMM will create a snapshot image each time the current model window is refreshed. To exit this mode, select **auto-snapshot** again to disable it. SIMM helps you to

remember by playing a snapshot sound each time a snapshot file is created.

In addition to these menu items, if the current model has any motions linked to it, then these motions will be listed at the bottom of the **snapshot** pop-up menu. Selecting one of these motions will cause the motion to be played back *starting at the current motion value* and using the current gear. A snapshot image will be created for each frame of motion until the end of the motion. To stop this process, you can click the **stop** button or select the motion's menu item in the **snapshot** pop-up menu again.

movie >

The movie command allows you to easily animations of the SIMM model window into AVI movie files. When you select the **movie** command, a pop-up menu is displayed. The menu contains the following items:

choose movie file...

This menu item displays a file browser dialog that allows you to specify the name and location of the movie file to be created by SIMM. If you do not use this option to specify a file name, SIMM will use a default name based on the name of the model, as described below.

use default movie file name

This menu item instructs SIMM to use a default name when creating a movie file. The default name is based on the name of the model and, if appropriate, the name of the motion. If you create a movie using the **auto-movie** option, the file name will be "<model name>.avi." If you create a movie of a motion, the file name will be "<model name> avi."

auto-movie

This menu item puts SIMM in *auto-movie* mode. In this mode, SIMM will add a frame to the movie each time the current model window is refreshed. To exit this mode, select **auto-movie** again to disable it.

In addition to these menu items, if the current model has any motions linked to it, then these motions will be listed at the bottom of the **movie** pop-up menu. Selecting one of these motions will cause the motion to be played back *starting at the current motion value* and using the current *motion speed*. Each image of the model showed on the screen will be written to the movie file. To stop this process, you can click the **stop** button or select the motion's menu item in the **movie** pop-up menu again.

Note: While SIMM is creating a movie of the model window, it display a message at the bottom of the window indicating that a recording is in progress. This message will not appear in the output movie file.

Note: SIMM uses a separate CPU thread to write model images to the movie file. Thus, if your computer has at least two CPUs, the animation of the motion will not be slowed by the writing of the movie file to disk. However, this also means that it may take up to a minute after the SIMM display has stopped before the movie file is completely written to disk. During this time, the movie file will be unavailable for viewing, and SIMM will not be able to start creating a another movie for the same model.

2.4.3 Forms

Generalized Coordinates Form

generalized coordinates

This form contains a numerical field for each generalized coordinate in the current model. You can use it to position the model accurately by typing in exact values for the generalized coordinates. Just click the left mouse button in the field of the generalized coordinate you want to change, then enter a new value. If the gencoord is clamped, you will not be allowed to input a value outside its range. If the gencoord

is not clamped, any values outside its range will be displayed in pink.

When you load a motion file, the motion variable (e.g., % of gait cycle) appears in this form as if it were a generalized coordinate. Thus, you can also position the model at a specific time step in a motion sequence by selecting the motion variable field and entering a value.

The **gencoord** command can be used to toggle the visibility of generalized coordinate fields. For models with more than 50 gencoords, all gencoord fields with be hidden by default to avoid filling the Model Viewer window with too much information.

Model Name Form

name

This field contains the name of the current model. This name appears in the title bar of the model window, and on the model icon when the model window is iconified. The model name can be changed by editing this field.

w, h (width, height)

These fields contain the width and height of the current model's window. If you need to create snapshot images of a particular resolution, you can enter the dimensions into these fields to set the model window to the same size as the desired snapshot image.

motion speed

This field controls the speed at which motions are animated when using the **start** or **movie** buttons. The value is the percentage of the actual speed of the motion, as represented by the *time* column in the motion data. You can enter any value between -1000.0 and 1000.0 into this field.

Note: If there is no column labeled *time* in the motion, SIMM will use the X column as time, and will assume that the data in this column represents time stamps in seconds. This may cause problems when animating the motion, espe-

cially if the data in the X column is not monotonically increasing. It is recommended that all motion files you load into SIMM include a column of time data. See Section 3.5.1, Motion File Header, for more information about specifying time and the X column in a motion file.

2.4.4 Sliders

Next to each field in the generalized coordinates form is a slider which can be used to change the value of the corresponding generalized coordinate. You can move the slider thumb to change the value quickly, or press the left and right arrow buttons to decrease or increase (respectively) the value a little at a time. The slider can not be used to set the gencoord value outside its range. If the gencoord value is outside its range, the slider thumb will display a red arrow to show this. If the current model has any motions linked to it, sliders for these motions are also located here.

Between each gencoord field and slider are two toggle buttons used to clamp (labeled 'C') and lock (labeled 'L') the gencoord. Gencoords can be clamped or locked from either the Model Viewer window or the Gencoord Editor window. If a gencoord is clamped, its value can never go outside its range. If, for example, a value outside the range is read in from a motion file, the gencoord value will be set to the closest limiting value (the range start or end). If a gencoord is unclamped, the value is allowed to go outside the range. If a gencoord is locked, its value cannot be changed by entering a new value in the form field or by moving the slider. The value can only be changed if the gencoord is restored (using the restore gencoord command in the Gencoord Editor) or the model's gencoord values are restored (using the restore gc values command in the Model Viewer).

2.4.5 Toggle Buttons

trackhall

This button toggles the trackball method of interactively translating and rotating the 3D model window view. See Section 2.4.6, Model Window Viewing Commands, for a description of trackball and non-trackball view navigation.

muscle points

This button toggles the display of larger muscle attachment points on the current model. When this option is off, the spheres representing the muscle points are displayed with the same radius and in the same color as the cylinders representing the muscle lines of action. When this option is on, the muscle points are displayed in blue and with larger spheres, making them easier to see and select for editing.

shadows

This button toggles the display of the body segment shadows. Initially, shadows of the bones are not displayed in the model window. When you turn shadows on, they are drawn only for the body segments for which you defined a shadow direction and distance in the segment definition. See Section 3.3.1, Body Segments, for more details on specifying the shadow parameters for a body segment.

crosshairs

This button toggles the display of cross hairs in the model window. Cross hairs are useful for locating the center of the model window, which is also the center of the perspective transformation that is applied to the model view.

2.4.6 Model-Window Viewing Commands

All of the commands described in this section can be used to change the view of a model without opening the Model Viewer window. Just activate the window of the model you want to move, and press the appropriate keys. The commands described at the beginning of this section can be used even if the Model Viewer icon in the heads-up display

(HUD) is not selected. The commands that do require the icon to be selected are described at the end, and are indicated as such.

There are two methods of changing the view of the model within the model window. The first method, which operates when the *trackball* toggle button in the Model Viewer window is turned on, works as follows. Pressing the Ctrl key and the left mouse button will pan the view within the model window. As you move the cursor in any direction, the model will follow. Pressing the Ctrl key and the middle mouse button lets you zoom in and out of the model. Moving the cursor to the right will zoom in, and moving it left will zoom out. Press the Ctrl key and the right mouse button to rotate the model with a trackball method. Imagine a trackball superimposed on the model window, and you can move it using the cursor. Moving the cursor in some direction will rotate the model about an axis perpendicular to that direction.

The second method of changing the model view is active when the trackball toggle button in the Model Viewer window is turned off. In this case the Ctrl key and mouse buttons function as follows. To rotate the model around the Xaxis of the world frame, which always points horizontally to the right, press the Ctrl key and the left mouse button. The direction and speed of rotation are determined by the position of the cursor in the model window. If the cursor is at the far right edge of the window, the model will rotate quickly in the positive direction around the X-axis. As you move the cursor left, towards the center of the window, the rate of rotation slows down. If the cursor is in the middle of the window, the rotation rate is zero. As you move the cursor further left, towards the left edge of the window, the rotation rate increases, but in the negative direction. In a similar fashion, pressing the Ctrl key and the middle mouse button rotates the entire model around the Y-axis, which always points straight up. Pressing the Ctrl key and the right mouse button rotates the model around the Z-axis, which points out of the screen towards you.

animating motions

If you have loaded a motion file (see Section 3.5, Motion Files) into SIMM, you can move the model according to the motion sequence as if the motion parameter (e.g., percent of gait cycle) were a gencoord. Just press the key[s] that you specified in the motion file and move the cursor horizontally to move the model forwards or backwards through the motion sequence. The keys associated with each motion are also shown in the Motion Editor window.

You can also animate a motion frame-by-frame using the left and right arrow keys. Press the left arrow key to move back a frame in the current hotkey motion, and the right arrow key to advance a frame. Pressing the up and down arrow keys cycles through the motions linked to the model, setting each one in turn to be the current hotkey motion.

changing display properties

When you right-click on a bone or world object in the model window, a pop-up menu is shown with various display options. You can change the object's drawmode and material, as well as turn on and off the display of the segment axes and object's normal vectors. You can also set the camera to "look at" that segment, as if you had used the **look at** command described in Section 2.4.2.

You can also right-click on many other model components to show a pop-up menu with display and editing commands. These commands are described in the editing tools specific to those model components.



The following commands work on the model only when the Model Viewer icon in the HUD is selected. When it is selected, the icon will be highlighted with a yellow frame, and the title bar of the Model Viewer will be highlighted in yellow. You can then use the following commands to change the view of the model and to articulate the joints.

- Pressing this key moves the model in towards you, making it larger.
- Pressing this key moves the model out away from you, making it smaller.
- Pressing this key moves the model to the left within the model window.
- Pressing this key moves the model to the right within the model window.
- Pressing this key moves the model up within the model window.
- Pressing this key moves the model down within the model window.

moving the joints

To move the joints of a model, you can press the keys that are listed in the definitions of the generalized coordinates (see Section 3.3.4, Generalized Coordinates). When you press these keys while the model window is the active window, the values of the generalized coordinates will change. For example, pressing the k key changes the value of *knee_angle* in the sample lower-extremity model and thus flexes and extends the knee. The speed and direction of the gencoord movements work the same way as the model rotations described above. If you keep the keys pressed to continue changing a gencoord, you will eventually reach the end of the gencoord's range. When this happens, the value of the gencoord will stop changing, and the model will stop

moving. However, when you move the cursor to the opposite side of the window, the gencoord will move in the other direction. If you specified that a gencoord can *wrap* around its range of motion, then you will not reach these stopping points. With wrapping, when a gencoord reaches the end of its range, it will wrap around to the other end of its range and continue to move.

RIGHT-CLICKING MODEL WINDOW BACKGROUND

If you right-click on the background in the model window (*i.e.*, away from any object), SIMM displays a pop-up menu with three options. Set all tools to this model sets all of the editing tools to that model. Edit model preferences displays a dialog box that lets you modify some of the model preferences. Look at segment "ground" sets the ground segment to be the fixed one for display purposes, as described in Section 2.4.2 in the description of the look at button.

Hot Keys

The following hot keys work in the model window regardless of which tool the heads-up display is set to.

- F2 Equivalent to the left mouse button.
- F3 Equivalent to the middle mouse button.
- *F4* Equivalent to the right mouse button.
- F5 Starts/stops the current (hotkey) motion. Equivalent to pressing the **start/stop** button in the Model Viewer.
- F6 Toggles the display of each of the muscles in the model. If they are all on they will be turned off, and vice versa. If some of the muscle are displayed, those will be turned off and the hidden ones will be turned on.

- F7 Toggles the display of the markers in the model. If all of the markers are on, pressing F7 will turn them all off. If some (or none) are on, they will all be turned on. This only affects the display of the [pink] markers defined in the body segments, not [experimental] markers in motion data.
- F10 Toggles the display of the selected polygons on bone1 (as specified in the Bone Editor).
- When you press and hold down the F11 key in the model window, a small box with a yellow border is shown in the bottom left of the window, containing information about the object currently under the tip of the cursor. Some of the objects will be highlighted as you mouse over them, while others will not. But every object will be annotated in the text box, including motion objects such as force plates, markers, and foot prints.
- F12 Saves an image of the model window to a TIFF file. Equivalent to pressing the **snapshot** button in the Model Viewer and choosing *take snapshot*.
- Pressing the right arrow key advances the current hotkey motion by one frame, and pressing the left arrow key moves it backward one frame. The hotkey motion can be selected using the up and down arrow keys, as described below.
- Pressing the up and down arrow keys cycles through the motions linked to the model, setting each one in turn to be the current hotkey motion. A message is then printed to the SIMM Messages window, indicating which motion is current. The motion "realtime connection" is always included in the list of motions that is cycled through, and refers to the motion that is animated when SIMM is connected to Cortex or EVaRT in real-time.

2.5 Plot Maker

The Plot Maker allows you to plot various computed properties of the muscle-tendon actuators, including muscle force, moment, moment arm, and length. You can create these plot curves for any set of muscles, as a function of any generalized coordinate or motion variable. In addition to generating data curves that are calculated from the musculo-skeletal model, there are also two ways you can read in and plot data that are stored in input files. The first method involves plot data files (see Section 3.6, Plot Files, for more information), and the second allows you to plot columns of data from motion files (see Section 3.5, Motion Files, for more information). The Plot Maker window is shown in Figure 2-2.

2.5.1 Selector Menu

model >

The model selector lets you choose which model you want to use to make plots. Since some of the plotting options depend on the model, each model has its own set of menus and forms to set the plotting options. When you select a new model with the model selector, the Plot Maker window is redrawn with the new model's menu settings. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

plot >

The plot selector lets you choose the plot to which you will add curves. Before you have made any plots, the pop-up menu contains one item, called *new*. Initially, *new* is the current plot, and it indicates that if you generate any curves, a new plot window will be created to display the curves. When you create a new plot window by this method, the

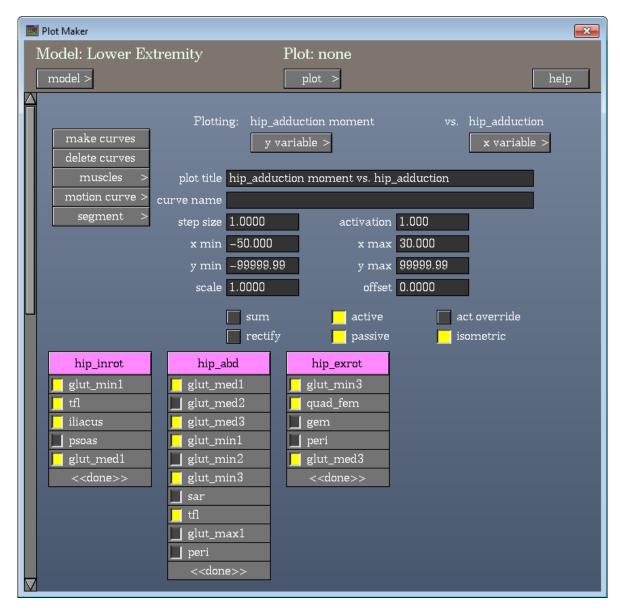


Figure 2-2. Plot Maker window

window is given a title (see below), and the current plot is changed to this new one.

help

Clicking on the help button opens a window containing helpful text about the Plot Maker. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.5.2 Command Menu

make curves

Select this command to create data curves from a model, according to the plotting parameters you have chosen (see Section 2.5.3, Generating Curves From a Model, for more details). The number of curves added to a plot when **make curves** is selected depends on the *sum* option (see Section 2.5.8, Toggle Buttons). If the *sum* option is off, then one curve will be added to the plot for each selected muscle. If the *sum* option is on, then the individual muscle curves will be summed, and the result (one curve) will be added to the plot.

delete curves

Selecting this command deletes all of the selected curves in the current plot window. To select or unselect a plot curve, click the left mouse button on the curve name in the plot key. This toggles the curve between being selected and unselected. When the curve is selected, the region around the curve name is highlighted in white; when unselected the region is black. You can also delete selected curves in a plot by pressing Backspace of Delete while the plot or plot key window is the active window.

muscles

>

This command lets you choose the muscles that will be used to generate plot curves. When you select this item, a pop-up menu of the muscle groups is displayed. Choosing a muscle group from this menu causes it to appear at the bottom of the Plot Maker window. If the chosen muscle group is already displayed at the bottom, selecting it again will turn

motion curve >

it off. You can click on individual muscles in any of the displayed group menus to select or deselect them for plotting.

This command lets you create plot curves from the data in motion files (see Section 3.5, Motion Files). When you select this command, a pop-up menu of motions is displayed. After choosing a motion, SIMM will display a window with checkboxes for all of the data columns in the motion. The items are organized into tabs such as Gencoords, Muscle Activations, and Force Plates. You can select any number of items to plot, although each plot in SIMM is limited to 60 curves. When you press *OK* the curves will be added to the current plot.

segment

This command lets you choose the body segment to be used with the *muscle orientation* Y-variable. For each selected muscle with an attachment point on this body segment, the vector describing the muscle line as it attaches to that body segment is defined as the muscle orientation. Consult Section 2.5.5, Y-variables, for more details.

2.5.3 Generating Curves from a Model

Use the following technique to plot computed properties (e.g., moment, force, length) of the muscles in a model. See Section 2.5.5, Y-variables, for a complete list of the musculotendon properties that you can plot.

- (1) If you have more than one model loaded into SIMM, use the model selector to make sure the current model is set correctly.
- (2) If there are already some plot windows on the screen, use the plot selector to choose which plot you want to add your new curves to. If you want to make a new plot window, choose the first item in the pop-up plot menu, labeled *new*. If there are no plot

- windows yet, the plot selector is already set to *new*, so you don't have to set it.
- (3) Select the Y-variable you want to plot (*e.g.*, force, moment) using the pop-up menu that is displayed when you click on **y-variable**.
- (4) Select the X-variable you want to plot against (a joint angle or motion variable) using the pop-up menu that is displayed when you click on **x-variable**.
- (5) Select the muscles whose properties you want to plot. Use the **muscles** command to display the appropriate muscle group menus, then choose the muscles from these group menus.
- (6) Set any other options, like *sum*, *active*, *passive*, and *curve name*.
- (7) Click on make curves.

Note: When plotting musculotendon properties that depend on activation (e.g., moment, force), make sure that the muscle activations are set correctly (they might have been changed by a motion variable). See the description of **act override** in Section 2.5.8, Toggle Buttons, for additional details.

2.5.4 Plotting Data from Motion Files

To plot columns of data from a motion file, you can use the **motion curve** command described in Section 2.5.2. This allows you to select one or more data columns and plot them versus the X column of the motion (usually "time").

You can also plot a column of motion data versus any other column of data in any motion. Here are the steps:

- (1) Click on **y-variable**. From the pop-up menu choose *motion curves*, then from the submenu choose the motion and then the data column that you want to plot on the Y-axis.
- (2) Click on **x-variable**. From the pop-up menu choose *motion curves*, then from the submenu choose the motion and then the data column that you want to plot on the X-axis.
- (3) Click on **make curves**. The Y data column will be plotted versus the X data column, and the curve will be added to the current plot.

2.5.5 Y-variables

moment

The moment of the selected muscles with respect to the chosen generalized coordinate. Moment is defined as force multiplied by the moment arm with respect to a generalized coordinate. The generalized coordinate is chosen from the submenu that appears when you move the cursor off the right side of the **moment** selection. You cannot compute the moment of a muscle with respect to a motion variable, so the **moment** submenu lists only true generalized coordinates. When selecting **moment** as the Y-variable, you must choose a generalized coordinate from the submenu.

musculotendon length

The origin-to-insertion length of the selected muscles. This length depends only on the locations of the muscle attachment points and the current configuration of the model's joints.

norm musculotendon length

The origin-to-insertion length of the selected muscles, divided by their respective lengths in a neutral position. By default the neutral position is the anatomical position (all gencoord values equal to zero). However, if the USE_DEFAULT_POSE_FOR_NORMALIZATION preference is set

musculotendon velocity

moment arm

moment arm (numerical) >

to *yes*, then instead of the anatomical position SIMM will use the default pose of the model. This Y-variable is useful when you want to compare muscle lengths between different subjects during an activity.

The contraction/lengthening velocity of the muscle-tendon actuator. When plotting versus a gencoord, the velocity will always be zero. When plotting versus a motion, the gencoord velocities from the motion data will be applied to the model in order to calculate musculotendon velocity.

The moment arm of the selected muscles with respect to the chosen generalized coordinate. The generalized coordinate is chosen from the moment arm submenu, which appears when you move the cursor off the right side of the **moment arm** selection. You cannot compute the moment arm of a muscle with respect to a motion variable, so the submenu lists only true generalized coordinates. The moment arm depends only on the locations of the muscle attachment points and the current configuration of the model's joints. It is calculated using the "partial-velocity" method described

This option is similar to the moment arm Y-variable described above, but uses a change-in-length technique to calculate the values. To calculate the moment arm for a given gencoord and body position, a central-difference technique is used with the muscle length calculations. The length of the muscle is calculated at the desired body position, and at positions corresponding to the gencoord value plus delta and the gencoord value minus delta. A quadratic function is fit to these three data points and the slope at the midpoint is used as the moment arm. This technique is useful if your model includes closed loops or constraint objects, which can interfere with the partial-velocity method of calculating moment arms.

fiber length

The fiber length of the selected muscles. To compute the fiber length of a muscle, SIMM finds the static equilibrium force in the muscle-tendon actuator. Thus the fiber length depends on the settings of the *active* and *passive* toggle buttons (see Section 2.5.8, Toggle Buttons), and also on the current activation level if the *active* component is turned on.

norm fiber length

The fiber length of the selected muscles, divided by their respective optimal fiber lengths. To compute the fiber length of a muscle, SIMM finds the static equilibrium force in the muscle-tendon actuator. Thus the fiber length depends on the settings of the *active* and *passive* toggle buttons (see Section 2.5.8, Toggle Buttons), and also on the current activation level if the *active* component is turned on.

force

The isometric force of the selected muscles. The force in a muscle depends on the settings of the *active* and *passive* toggle buttons (see Section 2.5.8, Toggle Buttons), and also on the current activation level if the *active* component is turned on.

moment@max force

The moment of the selected muscles calculated using the muscles' peak isometric force at all muscle lengths. Since this selection involves a moment calculation, similar to the **moment** selection described above, a generalized coordinate must be chosen. The generalized coordinate is chosen from the submenu that appears when you move the cursor off the right side of the **moment@max_force** selection. Since you cannot compute the moment of a muscle with respect to a motion variable, the submenu lists only true generalized coordinates.

tendon length

The tendon length of the selected muscles. To compute tendon length, SIMM finds the static-equilibrium force in the muscle-tendon actuator. Thus the tendon length depends on the settings of the *active* and *passive* toggle buttons (see

Section 2.5.8, Toggle Buttons), and also on the current activation level if the *active* component is turned on.

tendon strain

The tendon strain of the selected muscles. Tendon strain is defined as the stretch of the tendon divided by the tendon slack length. To compute tendon strain, SIMM finds the static-equilibrium force in the muscle-tendon actuator. Thus the tendon strain depends on the settings of the *active* and *passive* toggle buttons (see Section 2.5.8, Toggle Buttons), and also on the current activation level if the *active* component is turned on.

total positive moment

The sum of the moments of all the selected muscles, considering at each X value only those muscles that generate positive moment. That is, at each value in the selected generalized coordinate's range of motion, the moment of each selected muscle is calculated. All of the positive moments are then added together to get the total positive moment. If a muscle generates positive moment for some generalized coordinate values and negative moment for others, its moment will be included in the sum only when it is positive.

total negative moment

The sum of the moments of all the selected muscles, considering at each X value only those muscles that generate negative moment. That is, at each value in the selected generalized coordinate's range of motion, the moment of each selected muscle is calculated. All of the negative moments are then added together to get the total negative moment. If a muscle generates negative moment for some generalized coordinate values and positive moment for others, its moment will be included in the sum only when it is negative.

muscle orientation

This variable does not generate a plot curve, but rather writes out an ASCII data file with muscle orientation information. For each selected muscle with an attachment point

on the chosen body segment (the one selected using the **segment** command), the vector describing the muscle line as it attaches to that body segment is written to the data file. The point pt1 is the attachment point on the segment; the point pt2 is the next point on the muscle path, and vec is the normalized vector pt2 - pt1. This information is written to the file for each value of the corresponding X-variable. The name of the file is created using the names of the muscle[s], the name of the X-variable, and the suffix .mo. The folder that the file is created in is the current value of the OUTPUT_FILE_FOLDER preference.

motion curve

This option lets you plot one column of motion data versus another. The submenu contains a list of the motions linked to the current model, and each of those has submenus giving you access to all of the columns of motion data. To use this option you must select a motion curve as both the Y-variable and the X-variable. The selected curves do not have to be from the same motion, but they must contain the same number of data points. Once you have selected both, click on **make curves** to add the plot to the curve. The events for the motion will not be displayed in the plot since the X axis is not *time*.

2.5.6 X-variables

The possible independent (X) variables are the current model's generalized coordinates and the motion variables that are linked to the model. You can also select an individual column of motion data as the X-variable (see below). When you choose an X-variable, SIMM sets the *xmin* and *xmax* fields in the options form (see Section 2.5.7, Options Form) to the full range of motion for the generalized coordinate or motion variable. If you want to plot computed muscle properties over a smaller range, you should edit these form fields before generating your plot curves.

motion curve

This item in the X-variable pop-up menu lets you plot one column of motion data versus another. The submenu contains a list of the motions linked to the current model, and each of those has submenus giving you access to all of the columns of motion data. To use this option you must select a motion curve as both the Y-variable and the X-variable. The selected curves do not have to be from the same motion, but they must contain the same number of data points. Once you have selected both, click on **make curves** to add the plot to the curve. The events for the motion will not be displayed in the plot since the X axis is not *time*.

2.5.7 Options Form

plot title

This field contains the title of the current plot. If the current plot is *new*, then this field contains the title that will be used when a new plot window is created. The title of an existing plot can be changed by first making the plot the current one (using the plot selector), and then editing the plot title field.

When the current plot is *new*, the default-plot-title generator can be used to title the plot. If you have not yet edited the title of a *new* plot, the title defaults to a string containing the current Y-variable and the current X-variable. When you change the X and Y variables, the title changes to indicate the new variables. If you do not want to use the default title, you can edit the field. Once you enter your own title into the field, the default-plot-title generator is de-activated, so the title does not change when you choose different X and Y variables. If you want to turn the default-plot-title generator back on, edit the title field and enter a title of zero length (simply delete the current title). The title will then change to a default one consisting of the X and Y variable names, and it will then change as you select new X and Y variables as described above. Note that the default-plot-title generator functions only when the current plot is *new*. Once a plot has

been created, you cannot use the default-plot-title generator to automatically re-title it. As described above, however, you can enter your own title to an existing plot at any time.

curve name

This field contains the text string that will be used to name the next data curve that is generated. If the field is empty when a curve is created, a default name will be used. The default name depends on what you are plotting and the settings of the plotting options as follows.

When plotting computed muscle properties from your model, the curve name depends on the setting of the *sum* option. If *sum* is turned off, the name of each curve generated defaults to the name of the muscle used to create it. If the *sum* option is on, then the curve name defaults to a concatenation of muscle names used to make the curve.

When plotting data from motion files, the default curve name is the name of the motion followed by the label of the column of data as it appears in the motion file.

When plotting data from plot files, the default curve name is the name of the file. You can change this default either by entering the name of the curve directly into the plot file (see Section 3.6, Plot Files, for details), or by editing the *curve name* field.

If you enter your own curve name and then try to generate multiple curves at once (by selecting several muscles and turning off the *sum* option), the curve name will be deleted, and default names will be used for all the curves. When this happens, it usually indicates that you have forgotten to turn off the *sum* option or to select only one muscle for plotting.

To change the name of a plot curve after the curve has been created, right-click on the name in the plot key and choose *rename* from the pop-up menu.

step size

This field contains the step size used to generate data points for the plot curves. It affects how many data points make up a curve, and thus affects its resolution. For example, if you are making a plot curve for a generalized coordinate that ranges from 0 to 90 degrees, a step size of 30 will mean that the curve will have data points at 0, 30, 60, and 90 degrees. If you enter a step size greater than the range of the generalized coordinate (*e.g.*, 100 in the above example) the step size will be changed to the magnitude of the generalized coordinate range (90 in this case). When this happens, any curves generated will contain only two data points.

activation

This field contains a number between 0.0 and 1.0 representing the activation level of all of the muscles, as used for plotting purposes. For example, if this number is set to 1.0, then when you make a plot curve, all of the selected muscles will be assumed to be fully activated, regardless of their individual activation levels as listed in the muscle parameter menu (see Section 2.7, Muscle Editor). This "universal" activation level will be used only when *act override* is turned on (see Section 2.5.8, Toggle Buttons). Otherwise, the muscles' individual activation levels will be used.

x min. x max

These fields let you change the range of the generalized coordinate or motion variable which serves as the X-variable for making new data curves. When you choose an X-variable, the variable's minimum and maximum values are copied into the *x min* and *x max* fields, respectively. When you generate data curves, these values serve as the default range for the X-variable. You can change the range by editing these two fields. If you change the default range and then choose another X-variable, the values you entered are erased, and the new X-variable's default range is copied into the fields. If you then re-select the original X-variable, that variable's default range is copied into the fields, not the changed values you entered earlier.

y min, y max

These fields let you limit the range of the Y-coordinates of the data points in the curves that you add to a plot. By default, *y min* is a large negative number and *y max* is a large positive number, meaning that all of the data points in the curves should fall within the range and thus not get clipped. By changing these values, you can clip out data points that do not lie within a desired range. Every time a curve is added to a plot, each of the data points is compared to *y min* and *y max*. If the point is less than *y min*, it is set equal to *y min*. If it is greater than *y max*, it is set equal to *y max*. If the *sum* option is being used to sum curves from more than one selected muscle, *the individual curves are compared to this clipping range*, not the final summed curve.

scale

The scale field allows you to specify a scaling factor when generating plot curves using the **make curves** command. Whenever you generate a curve, each data point is multiplied by this scale factor before adding the curve to the plot window. This scaling is performed before the *offset* is applied.

offset

The offset field allows you to specify an offset factor when generating plot curves using the **make curves** command. Whenever you generate a curve, this offset value is added to each data point before adding the curve to the plot window. This offset is performed after the *scale* is applied.

2.5.8 Toggle Buttons

sum

This toggle button lets you choose whether or not you want to sum the curves of the selected muscles to generate a single curve. If *sum* is off, then each selected muscle creates a curve that is added to the plot. If *sum* is on, then the individual curves are summed and only one curve is added to the plot. The current setting of *sum* has no effect if you are plot-

ting data from plot data files or columns of data from motion files.

rectify

This toggle button controls the sign of the data points in the curves as they are added to a plot. If *rectify* is on, then all of the data points are made positive, in effect plotting the absolute value of the data. If it is turned off, then the data is not modified before plotting. Note that this option does not reverse the sign of the data points, but rather takes their absolute values, so be careful when using it while creating curves that have positive as well as negative data points. The current setting of *rectify* has no effect if you are plotting data from plot data files or columns of data from motion files.

active

This toggle button lets you choose whether or not to plot the active component of muscle force. Turning it on means that the force from the active force-length curve of the muscle will be included when computing muscle-tendon force. The current setting of *active* has no effect if you are plotting data from plot data files or columns of data from motion files.

passive

This toggle button lets you choose whether or not to plot the passive component of muscle force. Turning it on means that the force from the passive force-length curve of the muscle will be included when computing muscle-tendon force. The current setting of *passive* has no effect if you are plotting data from plot data files or columns of data from motion files.

act override

This toggle button controls the use of the "universal" activation level in the options form. When it is turned on, the activations of all of the muscles used for plotting will be set to the value shown in the *activation* field of the options form. This does not change their values in the muscle parameters form in the Muscle Editor; it merely sets the activations for plotting purposes only (to reset activations in the parameters

form to their defaults, see **reset activations** in Section 2.7.2, Command Menu). When act override is turned off, the muscles' individual activations as listed in the parameters form will be used. This toggle button is particularly useful when vou have loaded a motion file which contains muscle activations (see Section 3.5, Motion Files). After animating the model according to the motion, the muscle activations will be changed to match the values in the motion file (so that their size and color will vary during the animation). However, if you want to plot the muscles' maximum isometric forces during the motion, you can simply set the Plot Maker's activation level to 1.0 and turn on act override, rather than individually changing each muscle's activation in the Muscle Editor. There is no way to override just a subset of the muscle activations. The current setting of act override has no effect if you are plotting data from plot data files or columns of data from motion files.

isometric

This toggle button gives you control over the use of the force-velocity characteristics of the muscle fibers when calculating muscle force during a motion. When on, fiber velocities are set to zero and isometric muscle forces are calculated. When off, fiber velocity is used to scale the muscle force according to the muscle's force-velocity curve. See Section 3.4, Muscle Files, for details on how fiber velocities are estimated and used to scale isometric muscle force.

Note: fiber velocities affect only force and moment calculations, and only when plotting against motion variables which contain gencoord velocity information. For all other plotting scenarios, this toggle button has no effect.

2.6 Joint Editor

The Joint Editor enables you to graphically manipulate the kinematics of the joints. Once you select a joint, the three rotations and three translations (the "DOFs") that comprise the joint are displayed. If the rotations and translations are constants, you can change them by clicking in their fields and typing in new values. If they are functions of the generalized coordinates, they can be changed by moving, adding, and deleting the control points of the splines that define the kinematic functions (Figure 2-3). To display the kinematic function for a DOF, click on the field for that DOF. You can also change whether a given translation or rotation is a function or constant, thereby interactively increasing or decreasing the complexity of the joint model.

2.6.1 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Joint Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.6.2 Command Menu



This command lets you select the joint you want to edit. When a joint is selected, the six degrees of freedom (dofs) that comprise the joint are displayed in the dof form in the lower left corner of the Joint Editor window.



Figure 2-3. Joint Editor window

save current

This command saves the current joint of the current model. It saves the joint to a buffer so that you can restore it if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all the joints, so you can restore them back to their original form without having to save them first.

restore current

This command restores a previously saved joint description. Selecting it restores the current joint from a buffer that was saved last time you used the **save** command. When you first

create a model, SIMM saves copies of all of the joint descriptions, so you can restore a joint back to its original form without having to save it first.

2.6.3 Dof Form

r1 r2 r3 tx ty tz The dof form on the left of the Joint Editor window gives you access to each of the six degrees of freedom (dofs) in the current joint (see Section 3.3.3, Joints, for a description of the six dofs). If the dof is a constant, you can change the value by selecting its field with the left mouse button and then typing in a new value. You can change it to a function by right-clicking on the field and selecting *function* and then a generalized coordinate from the pop-up menu. If the dof is a function, you can right-click on the field and choose from the pop-up menu to change it to a constant, change the generalized coordinate used in the function, or send the function to the Function Editor for editing. If you change a dof from a constant to a function, SIMM creates a new function to use for the dof. This new function is given two control points, whose X-values are -360.0 and 360.0, and whose Yvalues are just below and just above the value of the dof when it was constant. If you change a dof from a function to a constant, its value is initially set to the average of the Ycoordinates of the function's control points. If you have closed loops in your model and the Inverse Kinematics solver is on, changing the complexity of a model will cause SIMM to re-evaluate the loop configurations. SIMM will try to find a configuration that closes all the loops. If no solution is found, an error message is displayed and the model is displayed with broken loops.

2.6.4 Transform Order

The transform order combobox allows you to change the order in which the dofs are processed when calculating the joint kinematics, which define the transformation from the parent reference frame to the child reference frame. Because the dofs are processed in a body-fixed manner (like an XYZ) Euler rotation), changing their order will change the joint transform. The three translations (tx, ty, tz) are always processed together, and are shown simply as "t" in the order. The order that is currently specified for the joint is shown in the combobox. When you left-click on it, a pop-up menu is displayed with all 24 possible orderings of the 4 elements. For most joints, you will want to specify the translation first. For example, when defining the hip joint between the pelvis and the femur, first you want to translate from the origin of the pelvis frame to the center of the femoral head. Then you want to rotate about that point using three orthogonal axes to define a ball-and-socket joint.

2.6.5 Rotation Axes Form

The form in the lower left corner of the Joint Editor window gives you access to the three rotation axes in the current joint. To display a rotation axis in the model window, click on the checkbox to the left of the axis defintion. To change the axis orientation, click on its X, Y, or Z field and type in a new value.

Note: When modifying a rotation axis, you do not need to enter the XYZ coordinates of a unit vector. SIMM will normalize the vector automatically when it calculates the joint transform. It will also normalize the vector and update the display of the coordinates in the Joint Editor when you choose a joint for editing. For example, you can enter values of 2.0, 2.0, 0.0 for a rotation axis and the Joint Editor will

show these values. When you switch the Joint Editor to a different joint and then back to the original one, the axis display will be updated to show 0.70711, 0.70711, 0.0. Also, these normalized values will be output when you save the model to a file.

2.7 Muscle Editor

The Muscle Editor gives you access to all of the parameters that describe the muscles. The paths of the muscles can be altered by selecting and moving the attachment points in the model window (see Sections 2.7.3 and 2.7.4). The Muscle Editor has several features to help you edit the muscle paths, such as an algorithm that moves a muscle point toward a specific vertex on a bone surface. The muscle-tendon parameters can also be changed with this tool (Figure 2-4). You cannot edit the force-length curves of the muscles or tendons with the Muscle Editor, but read Appendix A, Tips and Caveats, for details on how to edit these curves interactively in SIMM.

2.7.1 Selector Menu

model

The model selector lets you select which model's muscles you can change with the Muscle Editor. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Muscle Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

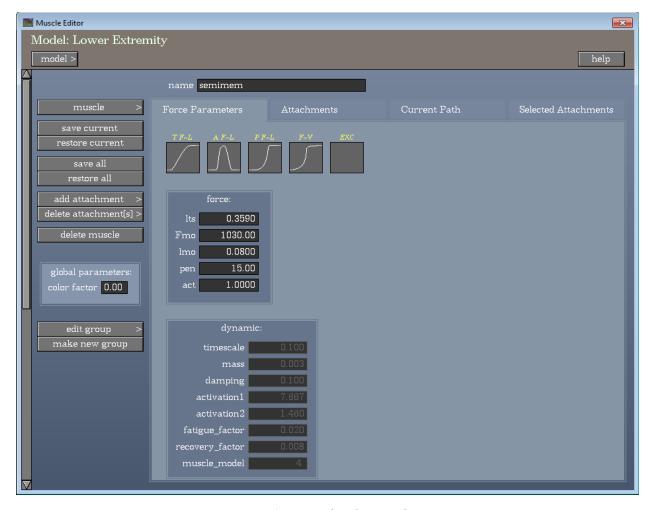


Figure 2-4. Muscle Editor window

2.7.2 Command Menu



This command lets you create a new muscle or choose an existing muscle for editing. When you select this item, a pop-up menu of existing muscles is displayed. The first item, *make new muscle*, creates a new muscle named

Muscle_X, adds it to the current model, and makes it the current muscle for editing. The muscle is attached to the first segment in the model and is given two arbitrarily positioned attachment points. The muscle properties are inherited from the default muscle, if there is one. If there is not, then selecting *make new muscle* will create a default muscle. The remaining items in the pop-up menu allow you to select existing muscles for editing.

save current

This command saves a copy of the current muscle in the current model. It saves it to a buffer so that you can restore it if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all of the muscles, so you can restore them back to their original form without having to save them first.

restore current

This command restores the current muscle in the current model from the buffer that was saved during the last **save all** or **save current** command. When you first load a model, SIMM saves copies of all of the muscles, so you can restore them back to their original form without saving them first.

save all

This command saves all of the muscles in the current model. It saves them to a buffer so that you can restore them if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all of the muscles, so you can restore them back to their original form without having to save them first.

restore all

This command restores all of the muscles in the current model from the buffer that was saved during the last **save all** or **save current** command. When you first load a model, SIMM saves copies of all of the muscles, so you can restore them back to their original form without saving them first.

add attachment

This command adds an attachment point to the current muscle. It displays a pop-up menu letting you choose where to add the new point in the list of existing points, and what body segment it will be attached to.

delete attachments >

This command lets you delete attachment points. It displays a pop-up menu with two primary options. You can delete an attachment point on the current muscle by selecting the first option (the name of the muscle) and then selecting a point from the submenu. You can delete all selected attachment points on all muscles in the current model by selecting the second option, *all selected attachments*.

delete muscle

This command deletes the currently selected muscle. The default muscle cannot be deleted.

edit group

This command allows you to edit the members of the muscle groups. Click on it to display a pop-up menu of the currently defined muscle groups. SIMM then displays a dialog box for adding or deleting individual muscles from the group.

make new group

This command allows you to create a new muscle group, which initially has no members. When you select it, SIMM displays a dialog box for specifying the name of the group. You are not allowed to specify the name of an existing group when making a new muscle group.

2.7.3 Global Parameter Commands

These commands control parameters that are not specific to a particular muscle.

color factor

This field contains the color factor, which is used to color all of the muscles according to their lengths. When the factor is 0.0, this technique is not used and the muscles are colored according to their material properties as specified in the muscle file. However, when the factor is greater than 0.0, the muscles are colored red when they are shorter than

their resting lengths, blue when they are longer than their resting lengths, and white when they are near their resting lengths. By changing the value of the color factor, you can change the transitions between blue, white, and red. A higher factor means that a muscle's color is more sensitive to length change, so that it will turn red more quickly when lengthening, and blue more quickly when shortening.

2.7.4 Force Parameters Panel

The Force Parameters panel gives you access to all of the parameters that govern the force-generating behavior of the muscles.

Muscle Force Functions

At the top of the panel are thumbnails for the five functions used in the force model. They are: the force-length curve of tendon (T F-L), the active force-length curve of the muscle fibers (A F-L), the passive force-length curve of the fibers (P F-L), the force-velocity curve of the fibers (F-V) and the excitation pattern of the muscle (EXC). When you right-click on one of the thumbnails, SIMM displays a pop-up menu with the following options.

inherit from default

This option is activated if the muscle is not currently inheriting the function from the default muscle. It changes the property so that it does use the same function that the default muscle uses. The function that it had been using is not deleted, even if it is no longer used in the model.

make unique

This option is activated if the function is currently being inherited from the default muscle. It makes a copy of the default function and uses it only for this muscle.

edit default function

This option is activated if the function is currently being inherited from the default muscle. It sends the function to the Function Editor so that it can be edited.

edit function

This option is activated if the function is not currently being inherited from the default muscle. It sends the function to the Function Editor so that it can be edited.

make new function

This option is activated only if there is no function for the force property. It creates a simple, two-point function that can then be edited in the Function Editor (after choosing right-clicking again and choosing *edit function*).

Force Parameters Form

On the left side of the Force Parameters panel is a form containing the five fiber and tendon parameters used by the SIMM mucle model. A detailed description of how these parameters are used to compute isometric muscle-tendon force is beyond the scope of this manual. For more details, consult the documents listed at the end of Section 1.6, SIMM Tutorials. To change the value of one of these parameters, click on its field and type in a new value. If the field is grayed out, it means that this parameter is inherited from the default muscle, and cannot be edited. If you want to modify the parameter for this muscle only, then you must first rightclick in the field and choose *make unique* from the pop-up menu. This will give the muscle its own copy of the parameter from the default muscle, which you can now edit by left-clicking on the field. To remove the muscle's own copy of the parameter and instead have it inherit it from the default muscle, choose *inherit from default* from the pop-up menu. If you want to edit the default muscle's copy of the parameter, you must first select the default muscle using the muscle button. Below is a description of the five parameters.

tendon slack length, the maximum length at which the tendon develops no force. This parameter is used to find tendon strain, which scales the X-coordinates of the tendon's force-length curve.

Fmo *maximum isometric force* that the muscle fibers can produce (active component). This parameter scales the Y-coordinates of the tendon's force-length curve and of the muscle's active and passive force-length curves.

Imo *optimal fiber length*, the fiber length at which maximum active isometric force is generated. This parameter scales the X-coordinates of the muscle's active and passive force-length curves.

pen pennation angle, the angle at which the muscle fibers insert on the tendon or bone. As pennation angle increases, the amount of muscle force that is transmitted through the tendon to the bone is decreased.

act level of activation, which can range from 0.0 to 1.0. This parameter scales the active component of isometric muscle force.

Dynamic Parameters Form

If the muscle has dynamic parameters defined (for use with the muscle models in Dynamics Pipeline simulations), SIMM will display them in a form on the right side of the Force Parameters panel labeled *dynamic*. If you do not plan to use this model for dynamic simulations, then you can ignore this form. You cannot add or delete dynamic parameters in the Muscle Editor, but you can change their values for the selected muscle via this form. To add or delete parameters, edit the muscle file as described in the Dynam-

ics Pipeline manual. To change the value of one of these parameters, click on its field and type in a new value. If the field is grayed out, it means that this parameter is inherited from the default muscle, and cannot be edited. If you want to modify the parameter for this muscle only, then you must first right-click in the field and choose *make unique* from the pop-up menu. This will give the muscle its own copy of the parameter from the default muscle, which you can now edit by left-clicking on the field. To remove the muscle's own copy of the parameter and instead have it inherit it from the default muscle, choose *inherit from default* from the pop-up menu. If you want to edit the default muscle's copy of the parameter, you must first select the default muscle using the **muscle** button.

2.7.5 Attachments Panel

This panel gives you access to all of the parameters that define the paths of the muscles. At the top of the panel are the fixed, via, and moving attachment points. The checkbox to the left of each attachment point allows you to select and deselect the point. To change the body segment to which a point is attached, click on the name of the current segment and choose a new one from the pop-up menu. For each X, Y, and Z coordinate of an attachment point, if the coordinate is a constant, you can change the value by selecting its field with the left mouse button and typing in a new value. You can change it to a function by right-clicking on the field and selecting function and then a generalized coordinate from the pop-up menu. If the coordinate is a function, you can right-click on the field and choose from the pop-up menu to change it to a constant, change the generalized coordinate used in the function, or send the function to the Function Editor for editing.

If the *Via* box is checked for the attachment, the point is a via point, meaning that it is active only when a generalized coordinate is in a certain range. For via points you can change the generalized coordinate that controls them by clicking on the name and choosing a different one from the pop-up menu. You can also change the coordinate range for which the point is active by clicking in the *Min* and *Max* fields and typing in a new value.



Figure 2-5. Muscle Editor: Attachments Panel

Muscle Wrapping Form

If the muscle has one or more wrap objects associated with it, SIMM will display the wrapping parameters in a region labeled *wrapping*. At the top of this region is a button labeled *object*. Clicking on it will display a pop-up menu of the wrap objects that are currently associated with the muscle. The order of the objects in the menu is the order in which they are processed when wrapping is calculated. If you want to change this order, you must use the Wrap Editor to unassign a wrap object from the muscle, and then reassign it [to the end of the list]. You can use this pop-up menu to choose which object's parameters you want to display and/or change.

start pt, end pt

These two fields are used to specify the region of the muscle path that is allowed to wrap over the wrap object. *Start pt*

defines the first muscle attachment point, and *end pt* defines the last attachment point in the region. Only muscle path segments that are between *start pt* and *end pt* will be checked for intersection with the wrap object. A value of -1 for *start pt* means to start at the first muscle point, and a value of -1 for *end pt* corresponds to the last point in the muscle. The muscle points are numbered starting at 1 and are in the order they are listed in the muscle definition, including all via points (points which are active only for certain gencoord ranges), whether or not the via points are active at the time wrapping is calculated.

hybrid, midpoint, axial

If the wrap object associated with the muscle is an ellipsoid, then you will also see a set of radio buttons, below the *start pt* and *end pt* fields, for specifying the desired ellipsoidal wrapping technique. The three techniques, hybird, midpoint, and axial, are described in detail in Section 2.10.5, *Ellipsoidal Wrapping Methods*. The method you choose with these radio buttons will be used for this muscle only. When you choose the wrapping method in the Wrap Editor, it will set the method for all muscles which are associated with that wrap object.

2.7.6 Current Path Panel

This panel shows the attachment points that define the path of the current muscle in the current body position. These points are determined by using the current generalized coordinate values to calculate the locations of the moving points, the on/off state of the via points, and whether the muscle wraps over any wrap objects. The list is updated in realtime as the body moves or as you edit the attachment points. The first three columns show the XYZ coordinates of the points with respect to the body segments to which they are attached. The next column shows the body segment, and the last column shows the type of attachment point.

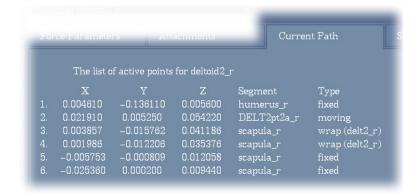


Figure 2-6. Muscle Editor: Current Path panel

When a muscle wraps over a wrap object, two attachment points are introduced to the list. One is the tangent point at which the path first comes into contact with the wrap object, and the other is the tangent point at which the path leaves the wrap object. The *Type* column also shows the name of the wrap object that is making contact.

2.7.7 Selected Attachments Panel

This panel lists all of the attachment points that are selected in the current model. The first column shows the name of the muscle followed by the index of the point (the same index as in the *Attachments* panel). The second column shows the type of point, and the third column shows the body segment to which the point is attached. If there is only one attachment point selected, and a polygon on that point's body segment is selected, then a fourth column is displayed in the panel, showing the name of the bone and the index of the polygon. The attachment point can be moved toward the vertices of the polygon as described in the next section.

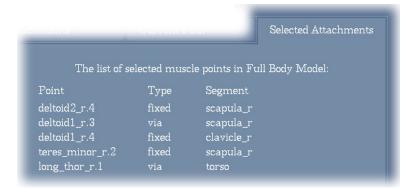


Figure 2-7. Muscle Editor: Selected Attachments panel

The selected attachment points shown in this panel are the ones that will move if you perform one of the editing commands described in Section 2.7.8, and they will be the ones deleted if you press the **delete attachment[s]** button and choose *all selected attachments*. It is often useful to see all of the selected points in one list so you can confirm that the proper ones are selected. Even if a selected point is not visible (e.g., when the muscle display is turned off or the point is hidden behind another object), it will still be moved or deleted with the other selected points.

2.7.8 Model-Window Commands

Muscle attachment points can be selected and edited using the *Attachments* panel described in Section 2.7.5. However, that panel operates only on the current muscle, while it is often desirable to modify attachment points of different muscles at the same time. The muscle editing commands in the model window, described below, allow you to do this.



Muscle attachment points can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Muscle Editor by clicking on the Muscle Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Muscle Editor, the icon will be highlighted with a yellow frame, and the title bar of the Muscle Editor will be highlighted in yellow.

SELECTING ATTACHMENT POINTS

To select muscle attachment points, press and hold the S S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting muscles points (N selected)," where N is the number of attachment points currently selected. As you move the cursor around the model window, if it hovers over an attachment point, the point will be highlighted with a yellow wireframe sphere, and the muscle name and point index will be shown in a tool tip box. If you then left-click on the point, it will be added to the group of selected points. If you click on an already selected point, it will be unselected. You can also box-select attachment points by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the points you want to select (or unselect). Box-select selects all muscle attachment points inside the box on all muscles currently being displayed, even if the muscle or point is hidden behind a bone or another muscle. This makes it possible to select a group of coincident points, such as those at the insertion of the Achilles tendon on the calcaneous.

As you select and deselect points, the list of currently selected points is updated in the *Selected Attachments* panel (if the Muscle Editor is set to the same model).

EDITING ATTACHMENT POINTS



To move the set of selected muscle attachment points, press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Moving muscle points: net move (seg) = X Y Z," where X, Y, and Z are the distances that the points have moved relative to the body segment seg. If all of the selected muscle points are attached to the same body segment, then seg is that segment. Otherwise, seg is the ground frame. To translate the selected points in the plane of the screen, press and hold down the left mouse button anywhere in the model window and move it to drag the points in that direction. A green arrow is also drawn in the model window starting at the original location of the first selected point, helping to show you how far the points have moved.



To move the selected attachment points in the X, Y, and Z directions relative to their respective body segments, use the x, y, z, and Shift keys. Pressing x translates the points in the negative X direction, while pressing Shift and x translates them in the positive X direction. Note: if multiple attachment points are selected and they are attached to different body segments, then the x, y, and z keys will move all of the points along the axes of the *ground* frame.

RIGHT-CLICKING ATTACHMENT POINTS

You can right-click on an attachment point to display a popup menu with several editing options. The options allow you to select/unselect the point for editing, create a new point before or after this point, delete the point, select the muscle for editing, hide the display of the muscle, hide the display of all muscles except this one, and delete the muscle. Note: you can right-click on any attachment point without holding down any keyboard keys, and without first clicking on the Muscle Editor icon in the HUD.

2.8 Gencoord Editor

The Gencoord Editor lets you view and edit properties of the generalized coordinates, such as their ranges and default values. It also lets you create and modify restraint functions for the gencoords, which are used by the inverse kinematics solver as well as by the Dynamics Pipeline.

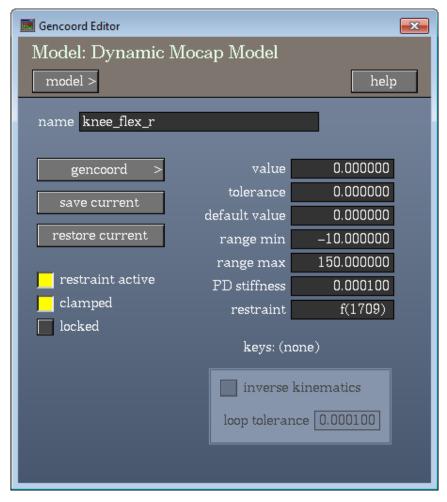


Figure 2-8. Gencoord Editor window

2.8.1 Selector Menu

model

The model selector lets you choose which model's gencoords to view and edit. When you choose a new model with the model selector, the Gencoord Editor window is redrawn with the settings of the new model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Gencoord Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.8.2 Command Menu

select gencoord

This command lets you select which gencoord's properties you want to edit. When a gencoord is selected, its properties can be viewed and edited using the toggle buttons and number fields in the tool window.

save current

This command saves the current gencoord of the current model. It saves all of the gencoord's properties to a buffer so that they can be restored if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all the gencoords, so you can restore them back to their original form without having to save them first.

restore current

This command restores a previously saved gencoord description. Selecting it restores the current gencoord's properties from a buffer that was saved last time you used the **save current** command. When you first create a model, SIMM saves copies of all of the gencoords, so you can restore a gencoord back to its original form without having to save it first.

2.8.3 Toggle Buttons

restraint active

This button allows you to toggle the restraint function of the current gencoord from active to inactive. When the restraint function is active, it can be edited and it will be used by the inverse kinematics solver if the gencoord is not clamped. If the restraint function is inactive, it is displayed in gray in the function plot area and is not used to restrain gencoord values when using the inverse kinematics solver. When the function is inactive, it cannot be edited, and points cannot be moved, added, or deleted.

clamped

This toggle button allows you to clamp or unclamp the current gencoord. If the gencoord is clamped, it is forced to remain within its specified range and cannot be set to values outside its range. This toggle button has the same effect as the toggle button in the Model Viewer, and is updated if the button in the Model Viewer is used. If the gencoord is unclamped, its value can go beyond its range, and it will be displayed in pink if it does so. If the gencoord is unclamped and has an active restraint function, this function will be used by the inverse kinematics solver. If a gencoord is outside its range when clamping is turned on, its value will be set to the closest range limit.

locked

This toggle button allows you to lock and unlock the current gencoord. When a gencoord is locked, its value cannot be changed by moving its slider, typing in a new value, playing back a motion, or by the inverse kinematics loop solver. Its value can be changed only by restoring the gencoord using the **restore current** command, or by applying a pose to the model, using the **restore pose** command in the Model Viewer.

2.8.4 Gencoord Form

The gencoord form in the lower left corner of the Gencoord Editor window contains numerical fields for the gencoord properties.

value

The current value of the gencoord is displayed in this field. If the gencoord is part of a loop, changing its value will cause the inverse kinematics solver to try to keep the loop closed by changing the other gencoords in the loop, while keeping the current gencoord constant. If it cannot find a solution at the current gencoord value, it will find the closest one. If the current gencoord is locked, its value cannot be changed by typing in a new value. If the gencoord is clamped, you cannot change its value to a number outside its specified range.

tolerance

The tolerance of a gencoord is used when the gencoord is set to a new value. If the difference between the new value and the current value is less than the tolerance, the new value is ignored, and the gencoord remains unchanged. This tolerance feature can be useful when you are importing a motion which contains identical or similar gencoord values for some periods of time. By setting the tolerance of each gencoord to, say, 0.5 degrees, small changes in gencoord values will not result in the joint transformation matrices being recalculated, thus saving some computation time. This feature is especially useful when you are importing a motion in real-time (with the Motion Module), and want the SIMM animation to be as fast as possible. The default value for tolerance is 0.0.

default value

The default value of a gencoord is the value that is reverts to when you select the **restore gc values** command in the Model Viewer. When you change the default value of a gencoord that is used in a loop, the inverse kinematics solver will re-evaluate the loop's default configuration, potentially

changing one or more gencoord's default values so that the loop remains closed. If default values of other gencoords in the loop are changed, a message window will pop-up to alert you to the new default values. You can then choose to (1) accept and save the new default values, (2) view but not save the values as defaults, or (3) discard them.

range min, range max

The range of the gencoord is displayed in the *range min* and *range max* fields. Both values can be changed by selecting the appropriate field with the left mouse button and then typing in a new value. Changing the range does not change the restraint function (if any), so you may need to edit the gencoord's restraint function after changing its range.

PD stiffness

This field holds a stiffness value that is used by the Dynamics Pipeline when calculating *gencoord moments (corr.)* while performing a dynamic simulation. This calculation uses a PD (proportional-derivative) controller to determine the moments necessary to drive a model to follow the input motion. Consult the Dynamics Pipeline manual for more information about this feaure.

restraint

This field shows which function is currently being used as the gencoord's restraint function (used during dynamic simulations and for inverse kinematics with closed-loop models when the gencoord is unclamped). If the gencoord has no restraint function you can right-click on this field and select *make new* from the pop-up menu. SIMM will create a new function based on the gencoord's range of motion, and use it as the restraint function. If this field shows the index of a function (*e.g.*, f(1709)), you can right-click on the field to display a pop-up menu with three options. Choose *edit* to send the function to the Function Editor for editing, choose *make new* to create a new function and use it as the gencoord's restraint function, and choose *delete* to remove the function as the gencoord restraint function (the function itself is not deleted).

keys

The *keys* display show you which keys control the current gencoord. See Section 2.4.6, Model Window Viewing Commands, for a description of how keyboard keys can be used to change a gencoord's value.

2.8.5 Inverse Kinematics

inverse kinematics

The inverse kinematics (IK) toggle button allows you to turn loop solving on and off. When the IK solver is on, SIMM will try to find a set of gencoord values that keeps all of the loops of body segments closed, while still satisfying the joint constraints. When solving, SIMM tries to keep clamped gencoords within their ranges, applies restraint functions (if specified) to unclamped gencoords to keep them within limits, and keeps locked gencoords at their locked values. When a model is loaded, SIMM uses the solver to assemble the system and close all the loops using the default values. If the solver is on when joint definitions are changed (changing dof types, editing functions), and when motions are applied to a model, the solver tries to close all loops given the current configuration. If individual gencoords are changed (by entering values, moving the slider, or using the gencoord keys in the model window), loops are solved keeping that gencoord value fixed (to the value you just set it to). Whenever the solver cannot find a solution with the exact gencoord value, it will try to find the solution with the closest gencoord value. In cases where no solution can be found, an error message will pop up and the model will be displayed with loops broken. If the restraint functions are poorly defined, the solver may be unable to find a solution because the errors are too large. An error message will be displayed alerting you of this. If there are no loops in your model, this button has no effect.

If your model has closed loops and constraints objects that involve the same gencoords, the IK solver will solve the loops and constraints simultaneously. That is, when a gencoord value is changed, or joint kinematics are modfied, the IK solver will adjust all of the other gencoords in order to enforce all loops and constraint objects at the same time. In some cases, having loops and constraint objects that involve the same gencoords may make finding a solution more difficult because there are more constraints on the gencoords that can be adjusted. If this problem arises, you may want to "soften" the constraint objects by adjusting the *weight* of the constraint points, as described in Section 2.13.6, Constraint Point Attributes.

Note: if you turn off the IK solver using the *inverse kinematics* toggle button, SIMM will still enforce the constraint objects.

loop_tolerance

This parameter specifies the "acceptance level" for the solutions that the solver module reaches when implementing the loop constraints. It does not affect the speed or accuracy of the loop solving. Rather, it is a cutoff value specifying whether or not the solution is acceptable. If it is not acceptable, a dialog box may be presented indicating that the loop constraint is not satisfied (this notice appears only when loading or editing a model, not when changing gencoord values). This tolerance value corresponds to the distance (in model units) between the two segments of the loop joint (see Section 3.3.3, Joints, for an explanation of the loop joint). When the loop is properly enforced, the points on the two segments are coincident, and the distance is zero. When the loop is not properly enforced, the loop joint disarticulates and the two points are separated by a certain distance. This parameter specifies how large a distance is considered acceptable. Its value can also be specified in the joint file, as described in Section 3.3.

2.9 Plot Viewer

The Plot Viewer lets you change the views of the plots. With this tool you can zoom in on a particular region of a plot curve, as well as display the coordinates of any point in a plot. The tool window available in previous versions of SIMM has been replaced with a pop-up menu that is displayed when you right-click in the plot window. Thus you can now access all of the Plot Viewer commands from within the plot window. The commands in the pop-up menu are described in the next section, and the other available commands are described in Section 2.9.2.

2.9.1 Pop-up Menu

delete points of interest

This command deletes all of the "points of interest" that have been added to the plot using the left mouse button, as described in Section 2.9.2.

delete lahels

This command deletes all of the labels that have been added to the plot using the *add label* menu item, described below.

reset view

This command resets the view of the plot to its initial view. The plot curves are zoomed and/or translated so that they are all visible within the window.

calculate mean of all curves

This command calculates the mean and standard deviation of all curves currently in the plot, and adds the results to the plot as a new curve. It is useful if you are plotting joint angles or muscle lengths for several subjects or motion trials and want to compute their averages. All of the curves in the plot must have the same X values in order to calculate their mean

show motion cursor hide motion cursor

This command toggles the display of the cursor in the current plot. The cursor is the vertical bar in the plot that shows

the current value of the motion variable that was used in creating the plot. The cursor is visible only if the plot is a function of some motion, and if the motion file header contains the "cursor" parameter (see Section 3.5, Motion Files, for details on specifying a plot cursor).

show events hide events

This command toggles the display of motion events. Motion events are vertical bars in plots that mark certain events in the motion that was used to create the plot. Motion events are visible only is the plot is a function of some motion, and if the motion file header contains one or more "event" parameters (see Section 3.5, Motion Files, for details on specifying a motion events).

show grid hide grid This command toggles the display of the grid lines drawn on the background of the plot.

show data points hide data points

This command toggles the display of the data points that comprise each plot curve.

show crosshairs hide crosshairs This command toggles the display of crosshairs that are drawn over the plot. The crosshairs track the motion of the cursor as you move it over the plot, with the XY coordinates of the intersection displayed as well. Also, if the crosshairs are close to a data point of one of the plot curves, the crosshairs will "lock onto" the data point. When this happens, the XY coordinates change color to match the curve color, and their values change to match the data point's XY values.

show Y=0

This command adjusts the Y range of the plot so that the horizontal axis (Y=0) is shown. This can be helpful if the plotted data has a small Y range far from the origin, and you want to get a better perspective on the magnitudes of the plot values.

show auxiliary data hide auxiliary data

This command toggles the display of the auxiliary data (if any) for the plot curves. Auxiliary data is specified in

motion files using the _aux suffix, and is commonly used to specify the on/off activation step function that corresponds to a channel of EMG data (the Bayes-filtered EMG is the column of motion data, and the step function is the auxiliary data). See Section 3.5.6, Auxiliary Data, for more details on specifying auxiliary data.

add lahel...

This command lets you add a text label to the plot, with the lower left corner of the text located at the cursor position when you right-clicked. Labels that are added to the data region of the plot will move with the data (e.g., when the plot is zoomed or scaled), and labels that are added to the margins of the data box will remain fixed in the margins.

save plot to file...

This command lets you save the plot to a file, and functions the same as the *Save Plot File* option in the *File* menu.

send to >

When you select this command, and then a model from the submenu, SIMM will copy the image of the plot window and display it on an object called *plotscreen.obj* in the model (you must add this object to the model before sending the plot to it). The plot in the model window will then update automatically as the regular plot changes. Similar to the *moviescreen.obj* object for displaying movies, *plotscreen.obj* can be dragged and dropped onto the model window, and then moved and scaled however as desired. A file named *plotscreen.obj* is available in the folder *SIMM\Resources\bones*.

2.9.2 Plot-Window Commands





These keys allow you to zoom and pan within the plot windows just like you can in the model windows. The i and o keys move the plot curves in and out, the u and d keys move the curves up and down, and the l and r keys move them left and right.



These keys allow you to zoom independently in the X and Y directions within the plot window. Pressing the x key will zoom in (in the X direction) on a part of the plot, and pressing the x key and the Shift key will zoom out (and similarly for the y key).

left mouse button

When you click the left mouse button in a plot window, a cross-hair is placed at the cursor location, and the coordinates of that point are displayed next to it. You can place up to 30 cross-hair points in each plot window. As you zoom in on a particular cross-hair using the i key, the precision of the displayed coordinates is increased so that you can more accurately note the point's location. To remove all cross-hair points from the plot, right-click in the plot window and choose the *delete points* option.

2.9.3 Live Plot Curves

A live plot curve is a plot of a muscle property that is updated automatically whenever any property of the muscle changes. Live plot curves are very useful when creating and modifying muscles because you can instantly observe the effects of moving an attachment point or a wrap object (or changing any other property) on the length, moment arm, or force of the muscle. To make a plot curve live, right-click on the curve name in the plot key, and choose *live* from the pop-up menu. A lower-case 'l' will be displayed on the left side of the key to indicate that the curve is live. Whenever any change is made to the model that affects the values in the plot curve, the curve is updated instantly and automatically. On slower computers, you may notice that the graphical interface will appear more sluggish if you make many plot curves live. It is recommended that you make plot curves live only when modifying muscle properties that will affect them. To stop a plot curve from being updated live, right-click on the name in the plot key and choose *live*.

2.10 Wrap Editor

The Wrap Editor allows you to create and edit muscle wrapping primitives (wrap objects). Wrap objects provide automatic muscle wrapping without the need to explicitly specify intermediate muscle points (a.k.a. via points). Muscle via points are described in Section 3.4.1, Muscles.

Each wrap object is associated with (fixed to) a specific body segment in the model, and its position is specified relative to the reference frame of that segment.

A wrap object may have one or more muscles assigned to it. When muscles assigned to a wrap object make contact with the wrap object's surface, the muscle is automatically deflected to prevent it from penetrating the wrap object.

A muscle may have more than one wrap object assigned to it. Each time a wrap object is assigned to a muscle, the object is added to the end of the muscle's list of wrap objects. When SIMM wraps a muscle over its assigned objects, the wrap objects are processed in the order in which they appear in this list. The resulting path of the muscle may depend on the order of processing, so care must be taken to define them in the appropriate order. To see the ordered list of wrap objects assigned to a given muscle, you must open the Muscle Editor tool and turn on the parameters for that muscle. The only way to change the order of the list is to unassign the wrap objects and then re-assign them in the desired order.

Wrap objects can take the form of spheres, cylinders, ellipsoids, or torii. Additionally, muscle wrapping can be constrained to a particular half of the wrapping primitive (*i.e.*, a hemisphere, hemicylinder, hemiellipsoid, or hemitorus).

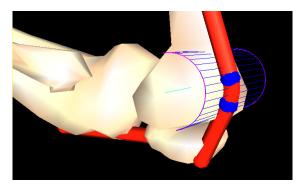


Figure 2-9. Cylindrical Muscle Wrap Example

2.10.1 Selector Menu

model >

The model selector lets you choose which model's wrap objects you wish to edit. When you select a new model with the model selector, the Wrap Editor window is redrawn with the new model's settings. If the current model has no wrap objects in it, then the Wrap Editor window will be blank except for the **wrap object** button which will let you add new wrap objects to the model.

help

Clicking on the help button opens a window containing helpful text about the Wrap Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.10.2 Command Menu

wrap object >

Select this command to create a new wrap object or to select an existing wrap object for editing. When you select this item a pop-up menu of existing wrap objects is displayed. The first item in the pop-up menu, *new wrap object*, creates a new wrap object named *Wrap_1*, adds it to the current model, and makes it the current wrap object for editing. The

remaining items in the pop-up menu allow you select existing wrap objects for editing.

segment

Select this command to change the body segment to which the current wrap object is attached (defined as the *parent* segment). When you select this item a pop-up menu of the model's segments is displayed. Choosing a segment from this menu will reassign the current wrap object to the selected segment. The wrap object's overall position will not change when it is assigned to a new segment. However, when the model is animated, the wrap object will follow the movement of its new parent segment.

muscles

This command lets you choose the set of muscles that wrap over the current wrap object. When you select this item, a pop-up menu of the muscle groups is displayed. Choosing a muscle group from this menu causes it to appear at the bottom of the Wrap Editor window. If the chosen muscle group is already displayed at the bottom of the window, then selecting it again will hide it. Selecting individual muscles from these group menus will cause the muscles to wrap over the current wrap object. The muscles will continue to wrap even if you turn off the display of these muscle menus.

save all

This command saves the current state of *all* wrap objects in the model to a buffer. It can be used in combination with the **restore** button to help you undo changes made while editing wrap objects.

restore all

This command restores the most recently saved copy of *all* of the current model's wrap objects. It can be used in combination with the **save** button to help you undo changes made while editing wrap objects.

delete

This command deletes the current wrap object from the model.

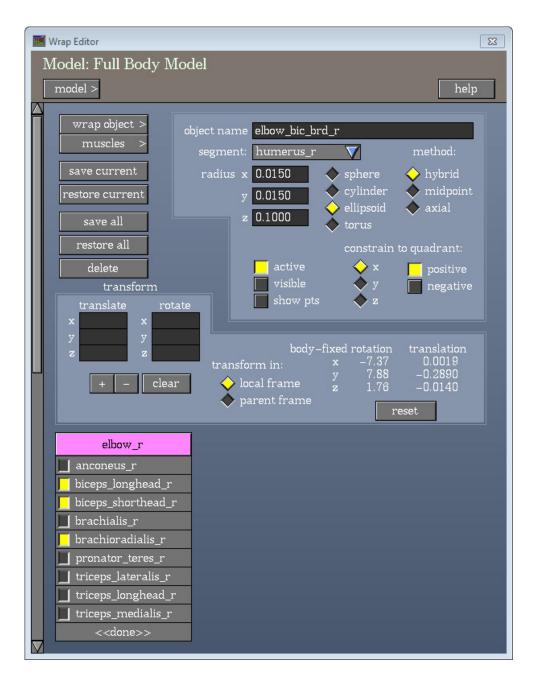


Figure 2-10. Wrap Editor window

2.10.3 Wrap Object Attributes

The panel to the right of the Wrap Editor's command menu can be used to edit the current wrap object's attributes. The previous section describes how to assign muscles to wrap objects and wrap objects to body segments. The controls described in this section allow you to edit the other parameters of the current wrap object.

object name

This field contains the current wrap object's name. Click and type in this field to rename the wrap object.

segment

This field displays the name of the parent segment of the current wrap object. To change the current wrap object's parent segment, click the **segment** button to the right of the field.

radius, height

These fields let you specify the size of the current wrap object. For spherical wrap objects there is only one field to specify: the sphere's radius. For cylinders there are two fields for the cylinder's radius and height. For ellipsoids there are three fields to specify the ellipsoid's X, Y, and Z radii. For torus objects there are two fields for the inner and outer radii.

sphere, cylinder, ellipsoid, torus

These radiobuttons let you choose the current wrap object's shape. Typically you will choose the shape that best resembles the surface you intend to wrap muscles over. Spheres and cylinders are simpler to define, and more straightforward for calculating wrapping. Ellipsoids are more complicated, and there are three techniques available for wrapping muscles over them (see Section 2.10.5 for a description of these methods). Torii are very useful as pulleys, but they work best if the muscle path remains roughly perpendicular to the plane of the torus.

hybrid, midpoint, axial

These radiobuttons let you choose which method is used to compute the wrapping path around ellipsoidal wrap objects. If the current wrap object is not an ellipsoid, then these radio-buttons will not be visible in the Wrap Editor window. See Section 2.10.5 for more information about the hybrid, midpoint, and axial wrapping methods.

active

This toggle button lets you turn the current wrapping object on and off. When a wrap object is not active, it is drawn in red, and it has no effect on muscles that intersect it.

visible

This button lets you toggle the visibility of the current wrap object in the model window. Once you are done editing a wrap object, you will typically make it invisible so that it does not obscure the display of other features of your model. A wrap object can be invisible, but still active.

show pts

This button toggles the visibility of muscle tangent points for muscles wrapping over the current wrap object. When checked, the X, Y, Z coordinates of the tangent points will be displayed in the model window.

constrain to quadrant

These radiobuttons and toggle buttons are used to specify that wrapping be constrained to a particular section of the current wrap object. The x, y, and z radiobuttons let you specify the constraint axis, and the *positive* and *negative* toggle buttons let you specify the constraint direction. For example, if the x radiobutton is selected with the *positive* toggle button, then wrapping will be constrained to the half of the wrap object in which all X-coordinates are greater than zero (in the object's local reference frame). When using these buttons, you may need to rotate the wrap object within its local reference frame in order to properly orient the active section of the wrap object.

2.10.4 Transforming Wrap Objects

The current wrap object can be moved, rotated, and scaled using the model window commands described in the next section, or by entering specific values into the numerical fields in the Wrap Editor window, as described below.

NUMERICAL WRAP OBJECT TRANSFORMATION

+, -, *clear*

To apply precise transformations to the current wrap object, enter the appropriate distances and angles into the *translate* and *rotate* fields, then click the + or - buttons one or more times to apply the transform or its inverse. If the *transform in local frame* radiobutton is selected then the specified transform will be applied relative to the wrap object's local XYZ axes. Otherwise the transform will be applied relative to the wrap object's parent segment's XYZ axes. The values you enter into these fields remain there after they are applied. To clear all values from the *translate* and *rotate* fields, click the **clear** button.

CURRENT TRANSFORM DISPLAY

transform in

The wrap object's current transform is displayed in the Wrap Editor window as a sequence of XYZ rotations and an XYZ translation. The order of the rotation sequence that is displayed is X, followed by Y, followed by Z. This order cannot be changed. If the *transform in local frame* radiobutton is selected, then the rotations displayed are about the wrap object's local XYZ axes (a.k.a. body-fixed rotation). If the *transform in parent frame* radiobutton is selected, then the rotation sequence displayed is about the wrap object's parent segment's XYZ axes (a.k.a. space-fixed rotation).

reset

Click the **reset** button below the current transform display to return the current wrap object to the origin of its parent segment's reference frame. The wrap object's transform is reset to the identity matrix.

2.10.5 Model-Window Commands



Wrap objects can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Wrap Editor by clicking on the Wrap Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Wrap Editor, the icon will be highlighted with a yellow frame, and the title bar of the Wrap Editor will be highlighted in yellow.

SELECTING WRAP OBJECTS

To select a wrap object, press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting wrap objects." As you move the cursor around the model window, if it hovers over a wrap object, the object will be highlighted and its name will be shown in a tool tip box (if the wrap object is already the current object in the Wrap Editor, it will be unhighlighted when you hover over it). If you then left-click on the wrap object, it will be selected and the Wrap Editor will update to show it as the current object. You cannot select more than one wrap object at a time.

EDITING WRAP OBJECTS

While this key is pressed, the HUD text box in the lower left corner of the model window displays "Editing current wrap object." Also, a GUI box will be displayed around the wrap object. This box lets you move, rotate, and scale the wrap object by left-clicking on the box handles. As you hover the cursor over a handle, a tool tip describes the handle's function. To move the wrap object, click on any edge of the box and drag it to a new location. To rotate the wrap object

about its local X, Y, or Z axis, click on one of the 6 circles on the box faces and spin it. To scale the wrap object, click on any of the GUI boxes at the corners of the box, the midpoints of the edges, or the centers of the faces and drag them. Note: not all GUI boxes will be displayed for all wrap object types. For example, spheres can only be scaled uniformly, so there are no edge or face handles, which would scale non-uniformly.

RIGHT-CLICKING WRAP OBJECTS

You can right-click on a wrap object to display a pop-up menu with several editing options. The options allow you to select the object for editing, activate/deactivate it, hide its display, hide all wrap objects other than this one, clone the object, and delete it. Note: you can right-click on any wrap object without holding down any keyboard keys, and without first clicking on the Wrap Editor icon in the HUD.

2.10.6 Ellipsoid Wrapping Methods

To overcome numerical instabilities involved with computing the optimal path over the surface of an ellipsoid, SIMM allows you to choose between three different algorithms for calculating muscle paths over ellipsoidal wrap objects. The default method, *hybrid*, works well for most muscles. However, if you notice erratic muscle motion (*e.g.*, skips or jumps when wrapping over an ellipsoidal wrap object) or your plots of muscle moment arms are not as smooth as you'd like, you should be able to achieve a smooth motion by experimenting with the *midpoint* or *axial* algorithms. In rare cases you may need to change the position or orientation of your wrap object to achieve smooth wrapping motion.

THE HYBRID WRAPPING METHOD

The hybrid method chooses a wrapping path by computing a weighted average of the results of the other two methods (described below). The accuracy of each result is used to determine its contribution to the overall path. In most cases this results in a smooth transition between the two wrapping methods. Occasionally you may experience an artificial change in slope in plots generated from muscles that use the hybrid wrapping method. This is a side effect of the hybrid method that can occur as the weighting shifts from favoring the midpoint result to favoring the axial result (or vice versa). In this situation you can use the wrap method radiobuttons in the Wrap Editor or Muscle Editor tools to restrict the choice of wrapping algorithm to either the midpoint or axial methods, thereby avoiding artifacts created as the hybrid method blends between the midpoint and axial methods. In most cases, however, the hybrid method will produce the best wrapping result.

THE MIDPOINT WRAPPING METHOD

The midpoint method chooses a wrapping plane (*i.e.*, the plane the muscle path will occupy when wrapped around the ellipsoid) by finding the point on the surface of the ellipsoid closest to the midpoint of the imaginary muscle line passing straight through the ellipsoid. This method produces smooth changes in the wrapping path unless the muscle line through the ellipsoid passes close to the center of the ellipsoid. In this situation the midpoint method becomes numerically unstable, creating erratic jumps in the wrapping path that it computes. If you are able to orient your wrap objects such that muscles do not pass near the ellipsoid's center, then the midpoint method will produce well-behaved wrapping paths.

THE AXIAL WRAPPING METHOD

The axial method chooses one of the ellipsoid's principal axes, X, Y, or Z, then computes a wrapping path based on the point where the imaginary muscle line passing straight through the ellipsoid intersects the plane perpendicular to that axis (i.e., the X=0, Y=0, or Z=0 planes). For example, if the X-axis is chosen, then the intersection of the muscle line with the x = 0 plane is used to find a point on the surface of the ellipsoid. The surface point is then used to determine the wrapping plane. The axial method works best when the muscle line is nearly parallel to the chosen axis. The accuracy of the axial method decreases as the angle between the muscle line and the chosen axis increases. Therefore the axial method automatically chooses the principal axis that is most parallel to the muscle line. This method produces a good result unless the "most parallel" axis happens to change during the range of motion of the muscle. In this situation a discontinuity in wrapping path will occur when the choice of axis switches from one principal axis to another. The hybrid method described above automatically detects when an axis switch is about to take place, and shifts its weighted average to favor the midpoint method to conceal the effect of the axis switch. If you are able to orient your wrap object such that muscles that wrap over it remain mostly parallel to the same axis throughout their range of motion, then the axial method will produce well-behaved wrapping paths.

2.10.7 Wrapping Tips & Techniques

Use the *constrain to quadrant* controls to specify which side of a wrap object muscles will wrap over. If you do not constrain wrapping to one side of a wrap object, then muscles will automatically wrap over the side that permits the shortest muscle path. Thus the muscle may switch from one side

of a wrap object to the other, depending on the positions of the body segments. By constraining the wrap object, you can force muscles to wrap over the desired side of the object even if it is not the shortest path around the object. However, when calculating a muscle path other than the true shortest path, SIMM may occasionally calculate an undesirable muscle path as it attempts to wrap over the active quadrants of the wrap object. If this occurs, you should rotate or translate the wrap object until the muscle paths behave as desired.

Cloning a wrap object can be useful if you are having trouble getting several muscles to wrap properly over a single wrap object. For example, if you already have a wrap object in place with two muscles wrapping over it, but would need to translate or rotate it slightly to properly wrap a third muscle, you can instead clone the object, adjust it as needed, and apply it only to the third muscle. To clone a wrap object, right-click on it in the model window and choose *clone* from the pop-up menu.

2.11 Deform Editor

The Deform Editor allows you to twist, bend, and warp portions of a body segment that would normally be considered rigid. When a segment is deformed, all bone vertices, distal joint centers, muscle attachment points, and muscle wrapping objects in the deformed region are transformed accordingly. In this way the effect of a bone deformity can be examined and measured using SIMM's other tools.

Modeling Bone Deformities in SIMM

Bone deformities are simulated in SIMM by assigning one or more deform objects to a body segment. A deform object is defined by two regular six-sided boxes, an inner box and an outer box. These two boxes are attached to a portion of the segment to determine which region of the segment will be deformed. Once the boxes are attached to the segment, the inner box may be freely transformed while the outer box remains stationary. As the inner box is moved or rotated, all bone vertices, distal joint centers, muscle points, and wrap objects associated with the segment that lie within the inner box follow its movement exactly. Elements of the body seg-

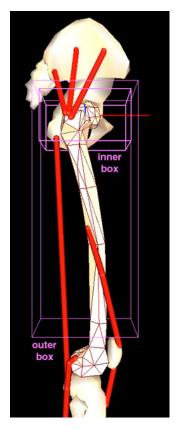


Figure 2-11. Attaching Deform Boxes

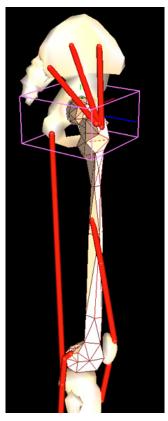


Figure 2-12. Applying Deformation

ment that lie between the walls of the inner and outer boxes experience a gradually decreasing amount of deformation. Elements outside the outer deform box remain unaffected.

Complex deformities can be modeled by attaching multiple deform objects to a body segment. Each deform object is responsible for deforming the particular region of the segment that it surrounds. Taken together the deform objects act in a sequence, the output of the first deform becoming the input of the second deform, and so on. Interactive transformation of multiple deform objects can be coordinated by defining a deformity object in the model's joint file. A deformity is simply a description of how to manipulate multiple deform objects at once. Each deformity has an associated slider added to the bottom of the Deform Editor window. Dragging a deformity's slider side-to-side causes all deform objects associated with the deformity to move through their range of motion at the same time.

2.11.1 Selector Menu

model :

The model selector lets you choose which model's deform objects you wish to edit. When you select a new model with the model selector, the Deform Editor window is redrawn with the new model's settings. If the current model has no deform objects in it, then the Deform Editor window will be blank except for the **segment** and **deform object** buttons which will let you add new deform objects to the model.

help

Clicking on the help button opens a window containing helpful text about the Deform Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

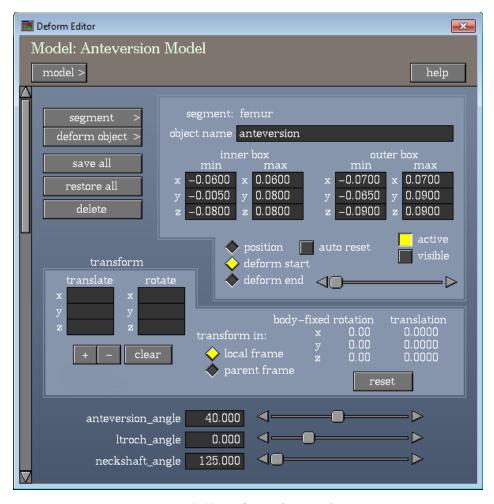


Figure 2-13. Deform Editor window

2.11.2 Command Menu



Select this command to specify the current segment for the Deform Editor. Any newly created deform objects will be added to this segment. The name of the current segment is displayed to the right of the button.

deform object

Select this command to create a new deform object or to choose an existing deform object for editing. When you click this button a pop-up menu of all deform objects assigned to the current segment is displayed. The first item in the pop-up menu, *new deform object*, creates a new deform named *Untitled*, adds it to the current body segment, and makes it the current deform object for editing. The remaining items in the pop-up menu allow you select existing deform objects for editing.

save

This command saves a snapshot of the current state of all deform objects in the model. It can be used in combination with the **restore** button to help you undo changes made while editing deform objects.

restore

This command restores the most recently saved snapshot of the current model's deform objects. It can be used in combination with the **save** button to help you undo changes made while editing deform objects.

delete

This command deletes the current deform object from the model.

2.11.3 Deform Object Attributes

The panel to the right of the Deform Editor command menu can be used to edit the current deform object's attributes.

segment

This field displays the name of the Deform Editor's current body segment. Any new deform objects will be added to this segment. You can use the segment button to the left of this field to change the current segment.

object name

This field contains the current deform object's name. Click and type in this field to rename the deform object.

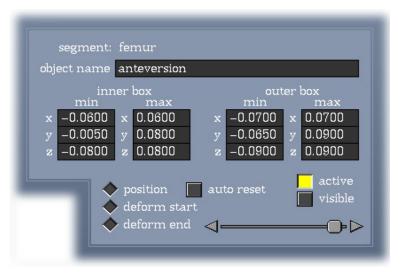


Figure 2-14. Deform Object Attributes Panel

inner box

These six fields contain the minimum and maximum X, Y, and Z dimensions of the deform object's inner box. Change these values to modify the size and shape of the current deform's inner box.

outer box

These six fields contain the minimum and maximum X, Y, and Z dimensions of the deform object's outer box. Change these values to modify the size and shape of the current deform's outer box.

position deform start deform end

These radio buttons specify the Deform Editor's current mode. In *position* mode you can move and rotate both the inner and outer deform boxes together to specify how they are attached to the current body segment. In *deform start* mode you can specify the inner box's starting transformation. In *deform end* mode you can specify the inner box's ending transformation.

The slider to the right of the *deform end* radiobutton allows you to smoothly move the inner deform box between its starting and ending positions.

auto-reset

This toggle button allows you to override the current deform object's standard behavior by making it automatically restore the current segment's origin to its original, undeformed, position. For more information on auto-reset deforms, see Section 2.11.5, Combining Multiple Deform Objects.

active

This toggle button lets you turn the current deform object on and off. When a deform object is deactivated it has no effect on bones, joints, muscle points, or wrap objects within it.

visible

This button lets you toggle the visibility of the current deform object in the model window. When the Deform Editor is in *position* mode both the inner and outer boxes are displayed for all visible deforms in the current segment. When the Deform Editor is in *deform start* or *deform end* modes only the inner box is displayed for all visible deforms in the current segment. Typically once you are done editing a deform object you will make it invisible so that it does not obscure the other features of your model.

2.11.4 Model-Window Commands



Deform objects can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Deform Editor by clicking on the Deform Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Deform Editor, the icon will be highlighted with a yellow frame, and the title bar of the Deform Editor will be highlighted in yellow.

SELECTING DEFORM OBJECTS

To select a deform object, press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting deform objects." As you move the cursor around the model window, if it hovers over a deform object, the object will be highlighted and its name will be shown in a tool tip box (if the deform object is already the current object in the Deform Editor, it will be unhighlighted when you hover over it). If you then left-click on the deform object, it will be selected and the Deform Editor will update to show it as the current object. You cannot select more than one deform object at a time.

EDITING DEFORM OBJECTS

To edit the current deform object, press and hold the E key. E While this key is pressed, the HUD text box in the lower left corner of the model window displays "Editing current deform object." After this message will be some text describing the type of editing you can perform on the object, as dictated by the current mode. Position mode is for specifying which portion of the body segment undergoes the deformation. In this mode you can translate the inner and outer boxes together by clicking on an edge of the outer box and dragging it. You can rotate the inner and outer boxes together by clicking on one of the circles on the outer box's faces and spinning it. Lastly, you can resize each box independently of the other by clicking on one of the corner, edge, or face handles and dragging it. Deform start mode is for specifying the starting position of the deformation. In this mode you can only translate and rotate the inner box. To do so, click on the box or one of the face circles and drag it. Deform end mode is for specifying the ending position of the deformation. In this mode you can only translate and rotate the inner box. To do so, click on the box or one of the face circles and drag it. As you hover the cursor over a handle in any of these deform modes, a tool tip describes the handle's function.

RIGHT-CLICKING DEFORM OBJECTS

You can right-click on a deform object to display a pop-up menu with several editing options. The options allow you to select the object for editing, activate/deactivate it, hide its display, hide all deform objects other than this one, and delete the object. Note: you can right-click on any deform object without holding down any keyboard keys, and without first clicking on the Deform Editor icon in the HUD.

2.11.5 Transforming Deform Objects

When adding deform objects to a model, most of your time will be spent choosing the proper size and placement for each object. The inner box and outer box fields described in Section 2.11.3 allow you to specify the size and shape of the current deform object. In addition to specifying the size and shape of your deform objects, you will need to transform them into position, and then transform them again to actually apply a deformation to the model.

The inner and outer boxes of the current deform object can be moved, rotated, and resized using the model window commands described in the previous section, or by entering specific values into the numerical fields in the Deform Editor window, as described below.

The collection of controls labeled *transform* can be used to position the current deform object within its parent segment's frame of reference (in *position* mode), and to apply a deform by moving and rotating the deform object's inner box (in *deform start* and *deform end* mode).

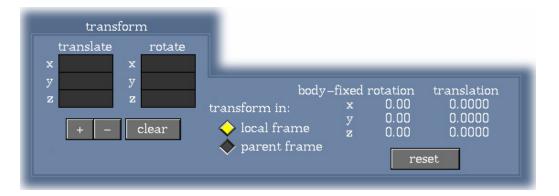


Figure 2-15. Deform Object Transform Panel

Numerical Deform Object Transformation

+, -, *clear*

To apply precise transformations to the current deform object, first enter the distances and angles to transform by in the *translate* and *rotate* fields, then click the + button below the text fields one or more times to apply the transform. If the *transform in local frame* radiobutton is selected then the specified transform will be applied relative to the deform object's local XYZ axes. Otherwise the transform will be applied relative to the deform object's parent segment's XYZ axes. To apply the inverse transform, click the - button. To clear all values from the translate and rotate fields, click the **clear** button.

Current Transform Display

transform in

The deform object's current transform is displayed in the lower right corner of the Deform Editor window as a sequence of XYZ rotations and an XYZ translation. The order of the rotation sequence that is displayed is X, followed by Y, followed by Z. This order cannot be changed. If the *transform in local frame* radiobutton is selected, then the rotations displayed are about the deform object's local XYZ

axes (a.k.a. body-fixed rotation). If the *transform in parent frame* radiobutton is selected, then the rotation sequence displayed is about the deform object's parent segment's XYZ axes (a.k.a. space-fixed rotation).

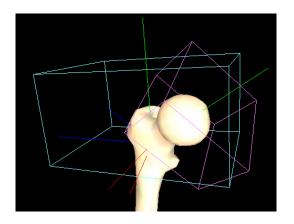
reset

Click the **reset** button to restore the current deform object's transform to identity. In *position* mode this will realign the deform object's reference frame with its associated body segment's reference frame. In *deform start* or *deform end* modes this will realign the deform object's inner box with its outer box.

Note: If you hold down the Alt or Ctrl key while clicking the *reset* button, each of the deformity sliders at the bottom of the Deform Editor window will be reset to their default values.

2.11.6 Combining Multiple Deform Objects

To simulate complex bone deformities you will typically need to apply more than one deform object to a body segment. Using multiple deform objects allows you to apply different types of deformation to different regions of the same segment. When working with multiple deform objects, it is important to keep in mind that the overall deformation applied to a body segment is the result of each of the individual deformations applied in sequence, starting with the first deform object in the segment's list. The order in which deform objects appear in their parent segment's list is significant. Rearranging a segment's list of deforms will change the aggregate deformation applied to the segment. Furthermore, the only way to rearrange a segment's list of deform objects is to edit the model's joint file, then reload the model. For these reasons it is helpful to have an understanding of how multiple deforms behave.



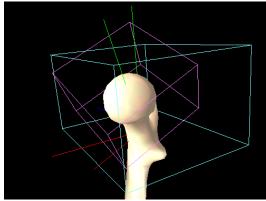


Figure 2-16. The left image shows the femur and deform boxes before any deformation has taken place. The right image shows the result of the first deform, represented by the larger box.

In Figure 2-14 above, the larger box in both images represents a deform object that twists the femur about its long axis. We call this deform object Anteversion. The smaller magenta box represents a deform object that bends neck of the femur up and down. We call this deform object Neck-Shaft Angle. In this example, Anteversion is the first deform in the list, followed by Neck-Shaft Angle. Notice in the picture on the right how the smaller Neck-Shaft Angle box has followed the twisting rotation of the larger *Anteversion* box. The reference frame of each deform box is itself deformed by any deform objects that precede it in the segment's list of deforms. This allows each deform box to "stick to" the region of bone that it was originally assigned to as deforms are applied to the segment. This also implies that the order in which deforms are applied contributes to the overall deformation of the body segment.

Auto-Reset Deform Objects

A special type of deform object called an auto-reset deform is available to simplify the task of maintaining the spatial characteristics of a deformed segment's proximal joint. Auto-reset deforms were designed specifically to keep the ball-and-socket joint at the hip intact as the femur undergoes deformation. However auto-reset deforms are not limited to the femur, they may be used to preserve the proximal joint of any deformed segment.

In general terms, an auto-reset deform ignores its *deform* start and deform end transformations and automatically computes a deforming transformation that restores the segment's origin to its original, undeformed position. If the segment's reference frame is defined such that its origin coincides with the center of its proximal joint, then the result of an auto-reset deform will be to restore the elements within its inner box to their original, undeformed position. For an auto-reset deform to behave correctly the following conditions must be met:

- (1) the origin of the body segment must coincide with the center of the segment's proximal joint,
- (2) the inner box of the auto-reset deform must contain the origin of the body segment, and
- (3) the auto-reset deform must be the last deform object in the segment's list of deforms.

In the hip joint example, the femur segment would be modeled such that its reference frame and proximal joint were centered within the femoral head. An auto-reset deform whose inner box surrounded the femoral head would be added to the end of the femur's deform list. The auto-reset deform's outer box would surround all of the femur except the distal joint. With this setup, the auto-rest deform would move and rotate the femoral head back into the acetabulum with its deforming effect gradually decreasing along the length of the femur. The knee joint would remain unaffected

by the auto-reset deform because it would lie outside of the auto-reset deform's outer box.

By default an auto-reset deform will apply both translation and rotation to restore the segment's origin to its original position and alignment. However, it is possible to specify that only translation be applied by including the translationonly keyword in the deform object's definition in the joint file. In this case the segment's origin is moved back to its original position, but the segment's reference frame may remain rotated from its original axes. There is no way to change the translation-only behavior of an auto-reset deform interactively within SIMM. See the Section 3.3.10, Deform Objects, for more information about how to specify the translationonly keyword in the joint file.

Using Deformity Sliders

Deformity sliders can be defined in a SIMM joint file as a way of coordinating the state of several deform objects with a single slider control. Each deformity in the joint file will appear as a slider at the bottom of the Deform Editor window.



Figure 2-17. Deformity Slider: Anteversion

The range and default value for a deformity are completely arbitrary and may be specified in the deformity's definition in the model's joint file. If the range is not specified it defaults to a minimum of 0.0 and a maximum of 100.0. If the default value is unspecified it defaults to 0.0. As you

drag a deformity slider side-to-side, each of the deform objects associated with the deformity will be adjusted such that their individual deform percentages match the deformity slider's percentage. Holding down the Alt or Ctrl keys and clicking the *reset* button just above the deformity sliders will automatically reset all deformity sliders to their default values.

In addition to controlling multiple deform objects, you may find it useful to set up deformity sliders to control individual deforms. This allows you to specify a meaningful range of values for a specific deform, control the deform's default value when the model is first loaded, and adjust the deform object's percentage without having to make it the current deform object.

To create or modify deformity sliders you must edit the model's joint file manually, then reload the model. There is no way to add, remove, or modify deformity sliders interactively within SIMM.

2.11.7 Deform Tips, Techniques, and Caveats

- The following list describes each element of a SIMM model that is deformed when it falls within the outer box of a deform object:
 - (1) bone surfaces: Bone vertices for all bones in the segment are deformed. However, bone vertex normals are not deformed. Therefore you may notice irregular shading of deformed bone surfaces
 - (2) distal joint centers: The origin of each joint emanating from the segment is deformed. However, the joint's DOFs and kinematic functions are not deformed.

- (3) *muscle attachment points*: Each muscle point attached to the segment is deformed.
- (4) *muscle wrapping objects*: The origin of each wrap object associated with the segment is deformed. However, the surface of the wrap object is not deformed. In other words, wrap objects are moved and rotated to follow a deformation, but the wrap object surface retains its shape (*i.e.*, sphere, cylinder, or ellipsoid).
- Keep in mind that deform objects are always associated with a specific body segment. A deform object will only affect elements that are also associated with that segment. Deformity sliders, on the other hand can control any combination of deform objects, and thus are not tied to a particular body segment.
- When moving muscle points, be careful not to move muscle points that are currently being deformed. Although the muscle point may appear to follow the motion of the mouse correctly, the next time you make a change to any deform object that is affecting the muscle point, the point will jump to a new location.

2.12 Marker Editor

The Marker Editor lets you edit the markers in your musculoskeletal model. Markers are used by the Motion Module to map motion capture data onto your model. If you do not use the Motion Module to import TRC or TRB motion files into SIMM, then you do not need to use markers or the Marker Editor. TRC and TRB files contain 3D coordinates for the markers that were placed on the subject while a motion was recorded. These files thus contain a series of

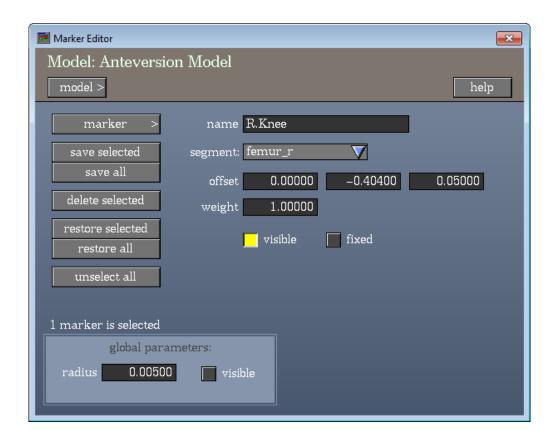


Figure 2-18. Marker Editor window

"marker clouds," each one representing the 3D coordinates of all the markers on the subject at a particular instant of time during the motion. The Motion Module reads these marker data and fits the SIMM model within the marker cloud for each time frame. By placing markers on the SIMM model that match the names and positions of the markers placed on the subject, the Motion Module is able to adjust the model's gencoord values to determine a "best fit" of the model to the marker cloud. The result is a SIMM motion that matches the TRC or TRB motion.

2.12.1 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Marker Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.12.2 Command Menu

marker >

This command lets you select the marker you want to edit. When you choose a marker from the pop-up menu, that marker is added to the set of selected markers; it does not unselect any previous selected markers. The parameters for the selected marker are displayed in the Marker Editor window. If more than one marker is selected, and they do not have the same parameter values (name, offset, weight), those fields in the tool window will be blank. To create a new marker, choose *make new marker* from the pop-up menu.

save selected

This command saves all of the selected markers in the current model. It saves them to a buffer so that you can restore them if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all the markers, so you can restore them back to their original form without having to save them first.

save all

This command saves all of the markers in the current model. It saves them to a buffer so that you can restore them if you make changes that you later want to undo. When you first load a model, SIMM saves copies of all the markers, so you

can restore them back to their original form without having to save them first.

delete selected

This command deletes all of the selected markers from the model.

restore selected

This command restores all of the selected markers in the current model. The markers are restored from a buffer that was saved last time you used the **save all** or **save selected** commands. When you first create a model, SIMM saves copies of all of the markers, so you can restore them back to their original form without having to save them first.

restore all

This command restores all of the markers in the current model. The markers are restored from a buffer that was saved last time you used the **save all** or **save selected** commands. When you first create a model, SIMM saves copies of all of the markers, so you can restore them back to their original form without having to save them first.

unselect all

This command unselects all of the selected markers in the current model.

2.12.3 Marker Parameters

name

This field shows the name of the selected marker. If more than one marker is selected, this field is blank and inactive. The names of the markers in your model should exactly match the names of the markers in the tracked marker files that you load into the model (although case does not matter). For models that are used as *mocap models*, the names of the markers must match the marker names in the marker sets that OrthoTrak recognizes. See Chapter 5, Motion Module, for more information on *mocap models* and the OrthoTrak marker set.

segment

This combo box lets you change the body segment to which a marker is fixed. When you change a marker's segment, the marker remains in the same position with respect to ground. The link between the marker and the origin of the segment it is fixed to updates to show the change.

offset

These three fields contain the X, Y, and Z offsets from the origin of the segment to the marker. If more than one marker is selected, and they do not have identical X, Y, or Z offsets, the corresponding fields will be blank. To change the offset, simply type in new values. If you enter an offset into a field that is blank because multiple markers are selected, the offset will be assigned to all selected markers. The units of the offsets are the implicit length units in your model (usually meters).

weight

This field contains the weight of the selected marker. If more than one marker is selected, and they do not have identical weights, this field will be blank. To change the weight, simply type in a new value. If you enter a weight into the field when it is blank because multiple markers are selected, the weight will be assigned to all selected markers. The weight of each marker determines how much emphasis is placed on it when the Motion Module fits the model to a frame of motion data. The higher the weight, the more the Motion Module tries to fit that marker to the data, at the expense of the other markers. If all the marker weights are the same (regardless of the actual weight value), the Motion Module treats all markers equally.

visible

This checkbox controls whether or not the marker is visible in the model window. Even if the marker is not visible, it is still used by the Motion Module when importing motion files.

fixed

This checkbox controls whether or not the marker is fixed. Fixed markers are optional markers whose X, Y, and Z off-

sets in the *mocap model* are not automatically calculated when the static trial is processed. Rather, the offsets in the marker definition in the joint file are used to position the marker on the model. These offsets are scaled with the body segment, however. Thus they are fixed to a certain region of the bone surface, even as the bone is scaled to fit the motion subject. The fixed / not fixed property of a marker is relevant only to the *mocap model*, since this property is used only when recalculating offsets during processing of the static trial.

2.12.4 Global Parameters

radius

This field contains the radius of the markers in the model. It affects only the display of the markers in the model window; it does not affect any marker offset calculations that the Motion Module performs when loading a *mocap model*.

visible

This checkbox controls the visibility of all of the markers in the model window. It is a convenience feature that allows you to turn all of the markers on or off without selecting them individually and changing their visibility. Even if the markers are not visible, they are still used by the Motion Module when importing tracked marker files.

212.5 Model-Window Commands

Markers can be selected and edited using the buttons and parameter fields in the Marker Editor window. However, the **marker** button only allows you to select one marker at a time, while it is often desirable to select and modify a set of markers at the same time (*e.g.*, all of the right ASIS markers). The muscle editing commands in the model window, described below, allow you to do this.



Markers can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Marker Editor by clicking on the Marker Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Marker Editor, the icon will be highlighted with a yellow frame, and the title bar of the Marker Editor will be highlighted in yellow.

SELECTING MARKERS



To select markers, press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting markers (N selected)," where N is the number of markers currently selected. As you move the cursor around the model window, if it hovers over a marker, the marker will be highlighted with a yellow wireframe sphere, and the marker name will be shown in a tool tip box. If you then left-click on the marker, it will be added to the group of selected markers. If you click on an already selected marker, it will be unselected. You can also box-select markers by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the markers you want to select (or unselect). Boxselect selects all markers inside the box whose visibility is turned on, even if the marker is hidden behind a bone or another marker. This makes it possible to select a group of coincident markers, such as those at a key anatomical landmark with different names.

As you select and deselect markers, the Marker Editor window updates to show the number of currently selected markers and their offsets and weights, which can be helpful for making sure only the appropriate ones are selected.

EDITING MARKERS

 (E)

To move the set of selected markers, press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Moving markers: net move (seg) = X Y Z," where X, Y, and Z are the distances that the markers have moved relative to the body segment seg. If all of the selected markers are attached to the same body segment, then seg is that segment. Otherwise, seg is the ground frame. To translate the selected markers in the plane of the screen, press and hold down the left mouse button anywhere in the model window and move it to drag the markers in that direction. A green arrow is also drawn in the model window starting at the original location of the first selected marker, helping to show you how far the markers have moved.

X Y Z
Shift

To move the selected markers in the X, Y, and Z directions relative to their respective body segments, you can use the x, y, z, and Shift keys. Pressing x translates the markers in the negative X direction, while pressing Shift and x translates them in the positive X direction. Note: if multiple markers are selected and they are attached to different body segments, then the x, y, and z keys will move all of the markers along the axes of the *ground* frame.

RIGHT-CLICKING MARKERS

You can right-click on a marker to display a pop-up menu with several editing options. The options allow you to select/unselect the marker for editing, hide the display of the marker, hide the display of all markers except this one, and delete the marker. Note: you can right-click on any marker without holding down any keyboard keys, and without first clicking on the Marker Editor icon in the HUD.

2.13 Constraint Editor

The Constraint Editor allows you to create and edit constraints in the model. These constraints provide a way of constraining points on one body segment to remain on the surface of a geometric primitive (a.k.a. constraint object) on a second segment. This mechanism is useful for creating joint kinematics that are too complex to be modeled with a traditional SIMM joint. For example, in the shoulder, two (or more) points on the scapula can be constrained to remain in contact with the ribcage, which is represented as an ellipsoid. Once this constraint is activated, when you change the value of one of the generalized coordinates in the shoulder, SIMM will adjust the values of the other generalized coordinates in order to satisy the constraints. This process is similar to the inverse kinematics algorithm that keeps loops together.

Each constraint is made up of a constraint object associated with a specific body segment in the model, and one or more constraint points on different body segments. A segment may have more than one constraint object attached to it. Constraint objects and points are defined within the reference frame of their associated segment and follow the movement of that segment. Constraint objects can take the form of planes, spheres, cylinders, or ellipsoids.

2.13.1 Selector Menu



The model selector lets you choose which model's constraint objects you wish to edit. When you select a new model with the model selector, the Constraint Editor window is redrawn with the new model's settings. If the current

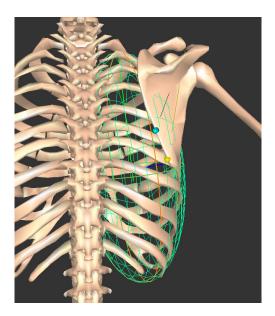


Figure 2-19. Sample Constraint. The constraint object is an ellipsoid, fixed to the ribcage. There are two constraint points, both fixed to the posterior edge of the scapula.

model has no constraint objects in it, then the Constraint Editor window will be blank except for the constraint **object** button which will let you add new constraint objects to the model

help

Clicking on the help button opens a window containing helpful text about the Constraint Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.13.2 **Objects Command Menu**

constraint object >

Select this command to create a new constraint object or to select an existing constraint object for editing. When you select this item a pop-up menu of existing constraint objects is displayed. The first item in the pop-up menu, new constraint object, creates a new constraint object, adds it to the current model, and makes it the current constraint object for editing. Object for editing. When a constraint object is created, a default point is created as well. The remaining items in the pop-up menu allow you select existing constraint objects for editing.

segment

Select this command to change the body segment to which the current constraint object is attached (defined as the parent segment). When you select this item a pop-up menu of the model's body segments is displayed. Choosing a segment from this menu will reassign the current constraint object to the selected segment. The constraint object's overall position will not change when it is assigned to a new segment. However, when the model is put in motion the constraint object will follow the movement of the new segment to which it is attached. Constraint objects and points cannot both be assigned to the same body segment, so in the pop-up menu the segment[s] on which the points are defined will be grayed out.

save all

This command saves the current state of *all* constraint objects in the model to a buffer. It can be used in combination with the **restore all** button to help you undo changes made while editing constraint objects.

restore all

This command restores the most recently saved copy of *all* of the current model's constraint objects. It can be used in combination with the **save all** button to help you undo changes made while editing constraint objects.

delete

This command deletes the current constraint object from the model.

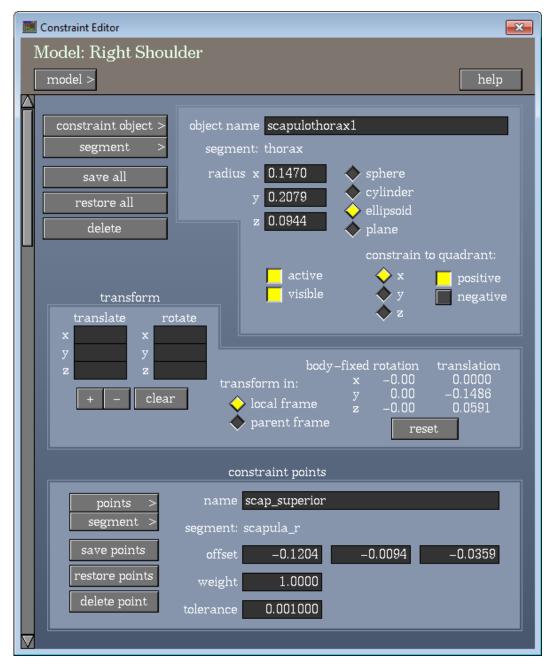


Figure 2-20. Constraint Editor window

2.13.3 Constraint Object Attributes

The panel to the right of the Constraint Editor's command menu can be used to edit the current constraint object's attributes.

object name

This field contains the current constraint object's name. Click and type in this field to rename the constraint object.

segment

This field displays the name of the parent segment of the current constraint object. To change the current constraint object's parent segment, click the **segment** button to the right of the field.

radius, height

These fields let you specify the size of the current constraint object. For spherical constraint objects there will be only one field to specify: the sphere's radius. For cylinders there will be two fields, for the cylinder's radius and height. For ellipsoids there will be three fields to specify the ellipsoid's X, Y, and Z radii. For planes there will be two fields to specify, the size of the plane in the X and Y directions (this is for display only; the plane is infinite for the purpose of implementing the constraint).

plane, sphere, cylinder, ellipsoid

These radiobuttons let you choose the current constraint object's shape. Typically you will choose the shape that most closely resembles the anatomical surface to which you want to constrain the points.

active

This toggle button lets you turn the current constraint object on and off. When a constraint object is not active, it is drawn in red, and SIMM does not enforce its constraints.

visible

This button toggles the visibility of the current constraint object in the 3D model window. Typically once you are done editing a constraint object you will make it invisible so that it does not obscure the display of other features of your model. A constraint object can be invisible but still active.

constrain to quadrant

These radiobuttons and toggle buttons are used to specify that constraint points be constrained to a particular half of the constraint object. The x, y, and z radiobuttons let you specify the constraint axis, and the positive and negative toggle buttons let you specify the constraint direction. For example, if the x radiobutton is selected with the positive toggle button, then the points would be constrained to the half of the constraint primitive in which all x-coordinates are greater than zero (in the constraint object's local reference frame).

2.13.4 Model-Window Commands



Constraint objects can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Constraint Editor by clicking on the Constraint Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Constraint Editor, the icon will be highlighted with a yellow frame, and the title bar of the Constraint Editor will be highlighted in yellow.

SELECTING CONSTRAINT OBJECTS

To select a constraint object, press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting constraint objects." As you move the cursor around the model window, if it hovers over a constraint object, the object will be highlighted and its name will be shown in a tool tip box (if the constraint object is already the current object in the Constraint Editor, it will be unhighlighted when you hover over it). If you then left-click on the constraint object, it will be selected and the Constraint Editor will update to show it as

the current object. You cannot select more than one constraint object at a time.

EDITING CONSTRAINT OBJECTS

(E)

To edit the current constraint object, press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Editing current constraint object." Also, a GUI box will be displayed around the constraint object. This box lets you move, rotate, and scale the constraint object by left-clicking on the box handles. As you hover the cursor over a handle, a tool tip describes the handle's function. To move the constraint object, click on any edge of the box and drag it to a new location. To rotate the constraint object about its local X, Y, or Z axis, click on one of the 6 circles on the box faces and spin it. To scale the constraint object, click on any of the GUI boxes at the corners of the box, the midpoints of the edges, or the centers of the faces and drag them. Note: not all GUI boxes will be displayed for all constraint object types. For example, spheres can only be scaled uniformly, so there are no edge or face handles, which would scale non-uniformly.

RIGHT-CLICKING CONSTRAINT OBJECTS

You can right-click on a constraint object to display a popup menu with several editing options. The options allow you to select the object for editing, activate/deactivate it, hide its display, hide all constraint objects other than this one, and delete the object. Note: you can right-click on any constraint object without holding down any keyboard keys, and without first clicking on the Constraint Editor icon in the HUD.

2.13.5 Transforming Constraint Objects

The collection of controls labeled *transform* can be used to move and rotate the current constraint object within its parent segment's frame of reference. The current constraint object's transformation can be modified interactively in the model window, or it can be modified precisely by entering values in the *translate* or *rotate* fields of the Constraint Editor window.

INTERACTIVE CONSTRAINT OBJECT TRANSFORMATION

The current constraint object can be moved, rotated, and scaled using the model window commands described in the previous section, or by entering specific values into the numerical fields in the Constraint Editor window, as described below.

NUMERICAL CONSTRAINT OBJECT TRANSFORMATION

+, -, clear

To apply precise transformations to the current constraint object, first enter the distances and angles to transform by in the *translate* and *rotate* fields, then click the + button below the text fields one or more times to apply the transform. If the *transform in local frame* radiobutton is selected then the specified transform will be applied relative to the constraint object's local XYZ axes. Otherwise the transform will be applied relative to the constraint object's parent segment's XYZ axes. To apply the inverse transform, click the - button. The values you enter into these fields remain there even after they are applied. To clear all values from the *translate* and *rotate* fields, click the **clear** button.

CURRENT TRANSFORM DISPLAY

transform in

The constraint object's current transform is displayed in the lower right corner of the Constraint Editor window as a

sequence of XYZ rotations and an XYZ translation. The order of the rotation sequence that is displayed is X, followed by Y, followed by Z. This order cannot be changed. If the *transform in local frame* radiobutton is selected, then the rotations displayed are about the constraint object's local XYZ axes (a.k.a. body-fixed rotation). If the *transform in parent frame* radiobutton is selected, then the rotation sequence displayed is about the constraint object's parent segment's XYZ axes (a.k.a. space-fixed rotation).

reset

Click the **reset** button below the current transform display to return the current constraint object to the origin of its parent segment's reference frame. The constraint object's transform is reset to the identity matrix.

2.13.6 Points Command Menu

constraint point >

When you select this item a pop-up menu of existing constraint points is displayed. The first item in the pop-up menu, new constraint point, creates a new constraint point, adds it to the current model, and makes it the current constraint point for editing. The remaining items in the pop-up menu allow you select existing constraint points for editing.

segment

Select this command to change the segment to which the current constraint point is attached (defined as the parent segment). When you select this item a pop-up menu of the model's segments is displayed. Choosing a segment from this menu will reassign the current constraint point to the selected segment. The constraint point's overall position will not change when it is assigned to a new segment. However, when the model is put in motion the constraint object will follow the movement of its new parent segment. Constraint objects and points cannot both be assigned to the same segment, so the parent segment of the associated constraint object will be grayed out.

save points

This command saves a snapshot of the current state of all constraint points for the current constraint object. It can be used in combination with the **restore points** button to help you undo changes made while editing constraint objects.

restore points

This command restores the most recently saved snapshot of the current constraint object's points. It can be used in combination with the **save points** button to help you undo changes made while editing constraint points.

delete point

This command deletes the current constraint point from the model.

2.13.7 Constraint Point Attributes

The panel to the right of the constraint point command menu can be used to edit the current point's attributes.

point name

This field contains the current constraint point's name. Click and type in this field to rename the constraint point.

segment

This field displays the name of the body segment to which the constraint point is attached. To change the current constraint point's segment, click the **segment** button to the right of the field.

offset

This field let you specify the location of the constraint point as an offset from the origin of its body segment.

weight

The *weight* of a point is a non-negative floating point value which controls how much emphasis is placed on it when enforcing the constraints. The higher the weight, the more SIMM tries to position that point onto the object surface, at the expense of the other points. If the weights of all of the points are the same (regardless of the actual weight value), SIMM will treat them all equally (*e.g.*, you cannot try to enforce a constraint better by increasing all the weights from 1.0 to 2.0).

tolerance

The *tolerance* of a point is the distance (in model units) that it is allowed to be from its correpsonding object, and still be an acceptable solution. It does not affect SIMM's calculation of the solution (as does the weight of a point). Rather, it is a threshold value which is checked after the best constraint solution has been calculated. If the point is not within *tolerance* of its constraint object, it is colored orange to indicate this condition.

2.14 Segment Editor

The Segment Editor allows you to edit the properties of the body segments in your SIMM model. Many of these properties are used only for Dynamics Pipeline simulations. If you will not be using your model for dynamic simulations, then the only relevant items in the Segment Editor are the display parameters such as drawing mode and material. If you will be performing dynamics, then this tool gives you access to the mass properties (mass, mass center, inertia), force matte definitions, and collision detection parameters (contact objects and spring-floors). Consult the Dynamics Pipeline manual for more information on how to use these parameters.

2.14.1 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Segment Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.



Figure 2-21. Segment Editor window

2.14.2 Command Menu

segment

>

This command lets you select the body segment you want to edit. When a segment is selected, the tool window is updated to show that segment's parameters.

save segment

This command saves the current state of the current segment in the model. It can be used in combination with the **restore segment** button to help you undo changes made while editing segments.

restore segment

This command restores the most recently saved copy of the model's current segment. It can be used in combination with the **save segment** button to help you undo changes made while editing segments.

drawmode

This command lets you change the drawing mode for all of the bones in the current segment. The six available modes are: gouraud shaded (filled polygons, smooth shading), flat shaded (filled polygons, flat shading), solid fill (filled polygons, no shading), outlined (white polygons with highlighted edges), wireframe (hollow polygons), and none (no display). To change the drawing mode of individual bones in a body segment, right-click on the bone in the model window.

material

This command lets you change the material that is used to display the bones in the current segment. The pop-up menu shows the materials that you defined in the joint file, plus a set of default materials that are defined within SIMM. To change the material of individual bones in a body segment, right-click on the bone in the model window.

2.14.3 Segment Parameters

Inertial Parameters

This panel can be used to modify the mass and inertia properties of the current segment. These properties are used only in dynamic simulations. The individual fields are described below

mass

This field lets you specify the mass of the current segment. You can specify the mass in whatever units you want, but they should match the units you use to specify inertias, muscle forces, and external forces and torques. The mass units in the models that come with SIMM are kilograms.

mass center

These fields let you specify the location of the mass center of the current segment. The coordinates are expressed in the body segment's reference frame.

inertia

These fields let you specify the inertia matrix of the current segment. These values are expressed in the reference frame of the body segment, and are relative to the mass center of the segment. You can specify inertia in whatever units you want, but they should match the units you use to specify mass, muscle forces, and external forces and torques. The inertia units in the models that come with SIMM are kilograms * meters * meters.

Force Matte

Force mattes are used only by dynamic simulations. They transform external forces expressed in the ground reference frame into the reference frame of the body segment to which the force matte is attached. Their primary use is to apply forces measured by forceplates to the feet during a simulation. To learn more about how force mattes work, consult the Dynamics Pipeline manual.

A body segment in SIMM can contain only one force matte. To create a force matte for the current segment, click on the **new force matte** button. The *name* field specifies the name of the force matte. The *filename* field specifies the name of the SIMM bone file which contains the polygon description[s] that comprise the force matte. Type a name into this field or click on the browse button to browse for a file. The *visible* checkbox controls whether or not the force matte is displayed in the model window. The *active* checkbox controls whether or not the force matte will be used in a dynamic simulation.

Contact Object

Contact objects are used by dynamic simulations to model rigid contact between polygonal surfaces. For more information on them, consult the Dynamics Pipeline manual.

The interface in the Segment Editor does not let you define all of the properties necessary to set up rigid-body contact detection for dynamic simulation, but it does let you define and view the polygonal contact objects. Once you have defined and positioned them appropriately in your model, you will need to edit the joint file and enter the contact pairs. Contact pairs specify which objects can contact which other objects.

To create a contact object for the current segment, click on the **contact object** button and choose *new* from the pop-up menu. The *name* field specifies the name of the contact object (which is used in the contact pair definitions). The *filename* field specifies the name of the SIMM bone file which contains the polyhedron that comprises the object. Type a name into this field or click on the browse button to browse for a file. The *visible* checkbox controls whether or not the contact object is displayed in the model window.

The *active* checkbox controls whether or not the contact object will be used in a dynamic simulation.

Spring Contact

Spring contacts are used only by dynamic simulations, and constitute a form of penalty-based contact detection. A spring contact consists of a *contact polyhedron* attached to one body segment, and one or more *spring points* attached to other body segments. When a point goes inside the polyhedron, a non-linear spring force is applied to the point's body segment to push it back out of the object. This technique is commonly used to model contact between the feet and the floor during a forward dynamic simulation. In this case, the contact polyhedron is a single polygon representing the floor, with a normal vector pointing upwards. Spring points are positioned on the bottoms of both feet; when they go below the floor, spring forces push them back up.

Each body segment can contain only one contact polyhedron. To create a spring contact for the current segment, click on the **new spring contact** button. The *name* field specifies the name of the spring contact. The *filename* field specifies the name of the SIMM bone file which containing the polyhedral object. Type a name into this field or click on the browse button to browse for a file. The *visible* checkbox controls whether or not the polyhedron and spring points are displayed in the model window. The *active* checkbox controls whether or not the spring contact will be used in a dynamic simulation. The *floor* checkbox, when checked, means that the contact polyhedron should be interpreted as a floor, with the first polygon in the file being used to define the plane of the floor. All other polygons in the file will be ignored.

Once a spring contact is defined, you can define one or more spring points on other body segments. Note that although these points are defined on segments other than the currently selected segment, they are only accessible in the Segment Editor when the current segment is the one to which their corresponding contact polyhedron is attached. The **spring point** button lets you select from the existing points or create a new one. The **segment** button lets you change the segment to which a point is attached. The X, Y, and Z fields specify the coordinates of the point in its associated segment's reference frame. The parameters *friction*, a, b, c, d, e, and f control the calculation of the spring force. Consult the Dynamics Pipeline manual for details on how these parameters are used.

NURBS Object

NURBS objects are used only by dynamic simulations, and they are used in conjunction with NURBS/spring objects to model contact between a NURBS surface and a mesh of springs. A NURBS object consists of an IGES file that contains the NURBS patches representing the surface of the object, and an information file (with suffix .nurbs) that contains scale, position, and surface normal information about the patches. You can make a NURBS object using the make NURBS obj button in the Bone Editor. To learn more about how NURBS-based contact detection works, consult the Dynamics Pipeline manual.

NURBS/spring

NURBS spring objects are used only by dynamic simulations, and they are used in conjunction with NURBS objects to model contact between a NURBS surface and a mesh of springs. A NURBS spring object consists of a spring information file (with suffix .spr) that contains position, normal, and stiffness information about the springs. You can make a spring object using the **make springs obj** button in the Bone Editor. To learn more about how NURBS-based con-

tact detection works, consult the Dynamics Pipeline manual.

The interface in the Segment Editor does not let you define all of the properties necessary to set up NURBS-based contact detection for dynamic simulation, but it does let you define and view the contact objects. Once you have defined and positioned them appropriately in your model, you will need to edit the joint file and enter the NURBS contact pairs. NURBS contact pairs specify which objects can contact which other objects.

Display Parameters

show mass center This toggle button lets you turn on and off the display of the

current segment's mass center location. If the mass center is

not specified, the button is deactivated.

show segment axes This button lets you toggle the visibility of the current seg-

ment's reference frame in the model window.

show normals This button lets you toggle the visibility of the normal vec-

tors for all of the bones in the current segment.

show inertia This toggle button lets you turn on and off the display of the

inertia vectors of the current segment.

Global Parameters

These display parameters are applied to all the segments in

the model.

show all mass centers This toggle button lets you turn on and off the display of the

mass center locations of all the segments in the model.

show inertia This toggle button lets you turn on and off the display of the

inertia vectors of all the segments in the model.

Scaling Parameters

The scaling parameters allow you to scale the current body segment along its local X, Y, and Z axes. You can type values into the X, Y, and Z fields, and then press **apply scale** to apply the new scale factors to the segment. Markers, muscle points, wrap objects, constraint objects, joint kinematics (for child segments) and all other components of a body segment are scaled along with the bones, *except for the inertial properties*. After the scale factors are applied to the segment, they are reset to 1.0.

2.14.4 Model-Window Commands



Body segments and spring points can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Segment Editor by clicking on the Segment Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Segment Editor, the icon will be highlighted with a yellow frame, and the title bar of the Segment Editor will be highlighted in yellow.

When the Segment Editor is the active tool in the model window, two buttons are displayed to the right of the Segment Editor icon in the HUD. These buttons are labeled *s* and *p*. When you click on the *s* button, the Segment Editor activates *segment* mode, meaning that you can select and edit body segments, as described below. When you click on the *p* button, the Segment Editor activates *spring point* mode, meaning that you can select and edit spring points, as described below.

SELECTING SEGMENTS



To select a body segment, first make sure the *s* button next to the Segment Editor icon in the HUD is highlighted, then press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting segment." As you move the cursor around the model window, if it hovers over a bone or any other model component that is fixed to one body segment, the name of that segment will be shown in a tool tip box. If you then left-click on the object, that body segment will be selected and the Segment Editor will update to show it as the current segment. You cannot select more than one segment at a time.

EDITING SEGMENTS



To edit the current segment, first make sure the *s* button next to the Segment Editor icon in the HUD is highlighted, then press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Editing segment." The only function currently supported for editing body segments is scaling. To scale the segment, press and hold the middle mouse button with the cursor anywhere in the model window. The HUD text box will update with "scaling" to confirm scaling mode. Also, a measurement bar is drawn over the model, starting at the origin of the segment and ending at the cursor. The current XYZ scale factors are shown at the end. Drag the cursor to increase or decrease the scaling factors, and release the middle mouse button when you are done.

SELECTING SPRING POINTS



To select spring points, first make sure the p button next to the Segment Editor icon in the HUD is highlighted, then press and hold the S key. While this key is pressed, the

HUD text box in the lower left corner of the model window displays "Selecting spring points." As you move the cursor around the model window, if it hovers over a spring point that is associated with the current body segment, the point will be highlighted with a yellow wireframe sphere, and the spring point name will be shown in a tool tip box. If you then left-click on the point, it will be added to the group of selected points. If you click on an already selected point, it will be unselected. You can also box-select spring points by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the points you want to select (or unselect).

As you select spring points, the Segment Editor window is updated to show the new parameter values. If more than one point is selected, and they do not have identical parameter values, those parameter fields in the tool window will be blank.

EDITING SPRING POINTS









To edit the current segment, first make sure the p button next to the Segment Editor icon in the HUD is highlighted, then press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Moving spring points: net move (seg) = X Y Z," where X, Y, and Z are the distances that the points have moved relative to the body segment seg. If all of the selected spring points are attached to the same body segment, then seg is that segment. Otherwise, seg is the current segment in the Segment Editor (the one containing the contact polyhedron). To translate the selected points in the plane of the screen, press and hold down the left mouse button anywhere in the model window and move it to drag the points in that direction. A green arrow is also drawn in the model window starting at the original location of the first

selected point, helping to show you how far the points have moved.



To move the selected spring points in the X, Y, and Z directions relative to their respective body segments, you can use the x, y, z, and Shift keys. Pressing x translates the points in the negative X direction, while pressing Shift and x translates them in the positive X direction. Note: if multiple spring points are selected and they are attached to different body segments, then the x, y, and z keys will move all of the points along the axes of the current segment in the Segment Editor (the one containing the contact polyhedron).

RIGHT-CLICKING SPRING POINTS

You can right-click on a spring point to display a pop-up menu with several editing options. The options allow you to select/unselect the point for editing, hide the display of the spring/object pair, deactivate the spring/object pair, and delete the point. Note: you can right-click on any spring point without holding down any keyboard keys, and without first clicking on the Segment Editor icon in the HUD.

2.15 Bone Editor

The Bone Editor allows you to transform bones (translate rotate, and scale) and move individual vertices of a bone, as well as perform simple boolean operations between bones (addition, subtraction, intersection). This tool is especially useful when you are creating a new model and need to move or scale bones within their reference frames. It provides a graphical interface to all the functionality of the *Norm* program, which is described in detail in Chapter 4.

2.15.1 Selector Menu

model

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Bone Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.15.2 Command Menu

Most of the operations in the command menu work on the bone specified by the **bone 1** button. This bone is called the current bone (which is not the same as the current segment). The boolean operations work on two bones specified by the **bone 1** and **bone 2** buttons.

do boolean

This command executes the specified boolean operation between *bone 1* and *bone 2*. The output bone is given the name that is in the *new bone name* field, and is stored in the current segment, which can be changed using the **change segment** button.

norm bone 1

This command runs *norm* on *bone 1*. Any transformations you have done to the bone are integrated into the bone file (the bone vertices are transformed), and the normals are recomputed. The translation and rotation fields are then reset to 0.0, and the scale fields are reset to 1.0.

split polygon

This command splits the selected polygon (displayed in green) by adding a vertex at the center of it and forming triangles using that vertex and each edge of the original polygon. This command works only if there is one selected polygon.

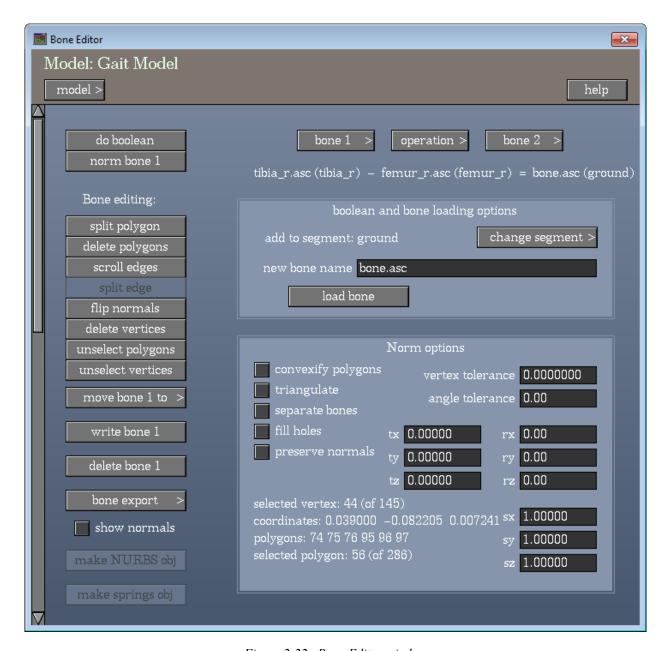


Figure 2-22. Bone Editor window

delete polygons

This command deletes the selected polygons. There is no way to undo this operation, except by reloading the bone file. However, in some cases you may be able to fill in holes created by deleting polygons by running *norm* on the bone with the *fill holes* option.

scroll edges

This command lets you change the selected edge (of the selected polygon) by scrolling through them. It works only if there is one selected polygon.

split edge

This command splits the selected edge in half, causing each of the two polygons which use it to be split into two pieces.

flip normals

This command reverses the directions of the normal vectors for the selected polygons. It then recomputes the vertex normals for the bone.

delete vertices

This command deletes all of the currently selected vertices.

unselect polygons

This command unselects all selected polygons on *bone 1*, including those that are hidden (using the F10 key).

move bone1 to

This command moves *bone 1* to the chosen body segment. It leaves the bone in its current position (relative to the rest of the model), but transforms the vertex coordinates so that they are expressed in the new segment's reference frame. It then runs *norm* on the bone using the current set of parameter values in the *norm options* region.

write bone 1

This command writes *bone 1* to a file. It displays a file browser to let you specify the name and location of the file.

delete bone

This command deletes *bone 1*. This operation cannot be undone.

bone export

This command allows you to export groups of bones to one or multiple files. There are three options you can choose from the pop-up menu:

all bones as OBJ files (same frames)

This option exports each bone in each body segment to its own OBJ file, keeping it in the reference frame of its body segment. This is an easy way to convert all of a model's bones to the OBJ format.

all bones to one file (frame "ground")

This option exports a single file (ASC or OBJ) containing every bone in the model. Using the current model pose, each bone is transformed into the ground reference frame and added to the file. This is an easy way to create a single bone file containing the model's skeleton in a certain pose.

all bones in <segment> to one file (frame "ground")

This option is available if *bone 1* is set to a valid bone. It exports a single file (ASC or OBJ) containing every bone in the body segment of *bone 1*. Using the current model pose, each bone in the segment is transformed into the ground reference frame and added to the file. This is an easy way to merge all of a segment's bones into a single file.

bone 1

This button lets you choose *bone 1*. *Bone 1* is the current bone, which can be transformed using the norm functions, edited with the polygon and vertex editing functions, and written to a file. *Bone 1* is also the first bone in the boolean operation, as shown in the boolean equation below this button.

operation

This button lets you choose the boolean operation to perform on *bone 1* and *bone 2*. The available operations are: *add*, *intersect*, and *subtract*. The result of the operation will be one or more polygonal surfaces, depending on the input. If the *separate bones* checkbox is turned off, then all of these surfaces will be combined into a single output bone. If the checkbox is off, each surface will be output to a separate

bone. The boolean operation is performed when you press the **do boolean** button.

bone 2 >

This button lets you choose *bone 2*, which is used only in boolean operations, as shown in the boolean equation below this button.

2.15.3 Boolean and Read/Write Options

Performing a boolean operation is similar to reading a bone file into SIMM in the sense that they both create a new bone. Thus the options to specify the name of the new bone, and to which body segment it is added, are located in the boolean and read/write options region.

change segment >

This command changes the body segment to which the new bone will be added, when a boolean operation is performed or when a bone file is read. It does not move the currently selected *bone 1* to that segment.

new bone name

This field specifies the name of the bone that will be created, either from the result of a boolean operation or from reading in a bone file.

load bone

This command loads a bone from a file. It displays a file browser allowing you to select the file, and it adds the bone to the body segment specified in the *add to segment* field.

2.15.4 Norm Options

Norm is used to process a bone when you select the **norm bone 1** button, and it is also used to postprocess the bone that is output from a boolean operation. The options to *norm* are used for both types of calls, unless otherwise specified.

When *norm* performs the specified transformations on a bone, it does them in the following order. First, it applies the rotations in a body-fixed manner (X, then Y, then Z). Then it applies the X, Y, Z scale factors and translations relative to the X, Y, and Z axes of the body segment's reference frame (not the rotated frame). After the transformations have been applied, the translation and rotation fields are reset to 0.0, and the scale fields are reset to 1.0. If you are unsure what the result will be of a series of different transformations, it is best to apply them one at a time, processing the bone with *norm* after each one.

When rotating, translating, or scaling a bone using either the keyboard commands (described in the next section) or typing values into the fields described below, remember to process the bone with *norm* (using the **norm bone 1** button) in order to apply the transformations to the bone vertices. Until you do this, the transformations are stored in a temporary buffer and the bone itself is unchanged (though you can see the effects of the transformation in the model window). When editing polygons and vertices of a bone, you should also run *norm* on it occasionally, to recompute the surface normals and update its shading. Lastly, you should frequently save the bone to a file as you make modifications, because there is no way to undo changes or restore the bone to a previously saved copy. If you make changes that you want to undo, you will need to delete the bone and read it back in from a previously saved file.

convexify polygons

When selected, *norm* will attempt to make every polygon in the bone convex by splitting concave polygons into convex pieces.

triangulate

When selected, *norm* will triangulate every polygon in the bone by splitting polygons (without adding vertices).

separate bones

When selected, *norm* will output a separate bone for each distinct polyhedral surface in the original object. This option is used only when postprocessing a bone created by a boolean operation.

fill holes

When selected, *norm* will attempt to fill in holes in the bone surface by adding triangles. If the bone contains large holes, *norm* may be unable to fill them.

preserve normals

When selected, *norm* will not recalculate polygon and vertex normals for the bone. This option is useful if you are working on a polyhedron that already has correct (or customized) normals that *norm* would modify (*e.g.*, because the polyhedron has holes or portions that are inside out).

vertex tolerance

The number in this field is the tolerance that norm uses to remove duplicate vertices. All vertices within *tolerance* of each other are merged into one vertex. A tolerance value of 0.0 will turn off vertex tolerance checking.

- tx, ty, tz These fields specify the X, Y, and Z translations of the bone within its segment's reference frame.
- rx, ry, rz These fields specify the X, Y, and Z rotations of the bone within its segment's reference frame. These rotations are applied in a body-fixed manner, in the order X, then Y, then Z (i.e., an Euler XYZ rotation). It is recommended that you apply one rotation at a time, so that the axis of each rotation remains parallel to one of the axes of the body segment's reference frame.
- sx, sy, sz These fields specify the X, Y, and Z scale factors of the bone, relative to its original size.

2.15.5 Model-Window Commands



Bones, polygons, NURBS patches, and polyhedron vertices can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Bone Editor by clicking on the Bone Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Bone Editor, the icon will be highlighted with a yellow frame, and the title bar of the Bone Editor will be highlighted in yellow.

When the Bone Editor is the active tool in the model window, three buttons are displayed to the right of the Bone Editor icon in the HUD. These buttons are labeled *b*, *p*, and *v*. When you click on the *b* button, the Bone Editor activates *bone* mode, meaning that you can select and edit bones, as described below. When you click on the *p* button, the Bone Editor activates *polygon/patch* mode, meaning that you can select and edit polygons (if the current bone is a polyhedron) or patches (if the current bone is a NURBS surface), as described below. When you click on the *v* button, the Bone Editor activates *vertex* mode, meaning that you can select and edit vertices (if the current bone is a polyhedron), as described below.

SELECTING BONES



To select a bone, first make sure the *b* button next to the Bone Editor icon in the HUD is highlighted, then press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting bone1." As you move the cursor around the model window, if it hovers over a bone, the name of the bone will be shown in a tool tip box. If you then left-click on the bone, it will be selected as *bone 1* and the Bone Edi-

tor will update accordingly. You cannot select more than one bone at a time.

EDITING BONES



To edit the current bone (bone 1), first make sure the b button next to the Bone Editor icon in the HUD is highlighted, then press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Editing bone1:" followed by some text describing the type of editing being performed. To translate the bone in the plane of the screen, press and hold the left mouse button anywhere in the model window. The HUD text box will update with "translating" to confirm translation mode. Also, a green arrow is also drawn in the model window starting at the original location of the bone's origin, helping to show you how far the bone has moved. Drag the cursor to move the bone in that direction, and release the left mouse button when you are done. To scale the bone uniformly in X, Y, and Z, press and hold the middle mouse button with the cursor anywhere in the model window. The HUD text box will update with "scaling" to confirm scaling mode. Also, a measurement bar is drawn over the model, starting at the origin of the bone and ending at the cursor. The current XYZ scale factors are shown at the end. Drag the cursor to increase or decrease the scaling factors, and release the middle mouse button when you are done. To rotate the bone about the segment's X, Y, and Z axes, press and hold the right mouse button with the cursor anywhere in the model window. The HUD text box will update with "rotating" to confirm rotation mode. Also, a circle is drawn over the model, centered on the origin of the bone, with the current XYZ rotation angles shown next to it. Dragging the cursor moves a virtual trackball that controls the rotation of the bone. Release the middle mouse button when you are done. As you transform

the bone with any of these editing commands, a "ghost" bone is drawn, showing its original position in the model.

(X) (Y) (Z)
(Shift)

You can also translate the bone along the X, Y, and Z axes of its body segment using the x, y, z, and shift keys. Pressing x translates the bone in the negative X direction, while pressing shift and x translates it in the positive X direction (similarly for Y and Z).

RIGHT-CLICKING BONES

You can right-click on a bone to display a pop-up menu with several display and editing options. The options allow you to change the drawmode and material of the bone, display its normal vectors, display the body segment's reference frame, center the viewing transformations on the segment, and select the bone or its body segment for editing. Note: you can right-click on any bone without holding down any keyboard keys, and without first clicking on the Bone Editor icon in the HUD.

SELECTING POLYGONS/PATCHES



If bone 1 is a polyhedron, you can use model-window commands to select and edit its polygons. If the bone is a NURBS surface, the same commands will select and edit the patches on the surface. To select polygons or patches, first make sure the p button next to the Bone Editor icon in the HUD is highlighted, then press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting polygons" or "Selecting patches." As you move the cursor around the model window, if it hovers over a polygon or patch on bone 1, the polygon/patch will be highlighted in blue, and its index will be shown in a tool tip box. If you then left-click on the polygon/patch, it will be highlighted in green and added to the group of selected objects. If you click on an

already selected polygon/patch, it will be unselected. You can also box-select polygons/patches by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the ones you want to select (or unselect).

EDITING POLYGONS/PATCHES



Selected polygons and patches cannot be edited using model-window commands. To delete or flip the normals of polygons and patches, use the buttons in the Bone Editor window. To translate polygons, select and edit the appropriate vertices, as described in the next section. Control points of NURBS surfaces cannot be edited.

SELECTING VERTICES



To select polygon vertices, first make sure the v button next to the Bone Editor icon in the HUD is highlighted, then press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting vertices." As you move the cursor around the model window, if it hovers over a vertex on bone 1, the vertex will be highlighted with a light-blue wireframe sphere, and the vertex index will be shown in a tool tip box. If you then left-click on the vertex, it will be highlighted with a red sphere and added to the group of selected vertices. If you click on an already selected vertex, it will be unselected. You can also box-select vertices by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the vertices you want to select (or unselect). You cannot select the control points of a NURBS surface.

EDITING VERTICES



To edit the selected polygon vertices, first make sure the ν button next to the Bone Editor icon in the HUD is highlighted, then press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Moving vertices: net move = XY Z," where X, Y, and Z are the distances that the points have moved. To translate the vertices in the plane of the screen, press and hold down the left mouse button anywhere in the model window and move it to drag the vertices in that direction. A green arrow is also drawn in the model window starting at the original location of the first selected vertex, helping to show you how far the vertices have moved.



To move the selected vertices in the X, Y, and Z directions relative to the bone's current reference frame, you can use the x, y, z, and \mathtt{Shift} keys. Pressing x translates the points in the negative X direction, while pressing \mathtt{Shift} and x translates them in the positive X direction. Note: the X, Y, and Z axes used for this command are the current axes of the bone. If there are non-zero rotations applied to the bone, the bone's axes will not be the same as the body segment's axes. To make them the same, process the bone with *norm* to apply (and then reset) the current transform.

2.16 Dynamics Tool

The Dynamics Tool is a graphical interface for running dynamic simulations created by the Dynamics Pipeline module. These simulations are compiled as DLLs (dynamic link libraries), and can be loaded into SIMM for execution. The Dynamics Tool allows you to set various parameters of the simulation, such as the input motion (for inverse dynam-

ics) and the desired output variables (e.g., moments, powers).

The Dynamics Pipeline performs forward and inverse dynamics analyses using the same numerical method. In both cases, the equations of motion are integrated forward in time using a variable-step integrator. Forces and torques are applied to the body segments (from gravity, muscles, ligaments, ground reactions, etc.), and the motion of the joints is calculated. If the motion of some or all of the joints is prescribed (e.g., to follow motion capture data), then the dynamics code calculates and applies the necessary constraint torques in order to ensure that the joints move as prescribed. In a typical forward dynamics simulation, muscle and external forces are applied, and the motion of the joints is not prescribed. In a typical inverse simulation, groundreaction forces are applied (but not muscle forces), and the motion of all of the joints is prescribed. In this case, the simulation will calculate the torques that are required at each joint to create the prescribed motion. However, because the same numerical method is used to perform both simulations. it is possible to mix forward and inverse dynamics. For example, you can prescribe the motion of all of the joints, but also specify the excitations of some of the muscles (e.g., from EMG recordings). In this case, the simulation will apply the muscle forces to the body segments, and then calculate and apply any additional torques that are required to prescribe the motion. This scenario is useful during an inverse dynamics analysis if you want to estimate the forces in the muscles whose EMG data you are collecting.

Dynamics Pipeline users can create their own DLLs of whatever musculoskeletal models they want, and thus can perform dynamic simulations on them in SIMM. FIT Module users cannot create their own DLLs, but can choose from several pre-built DLLs that come with the software.

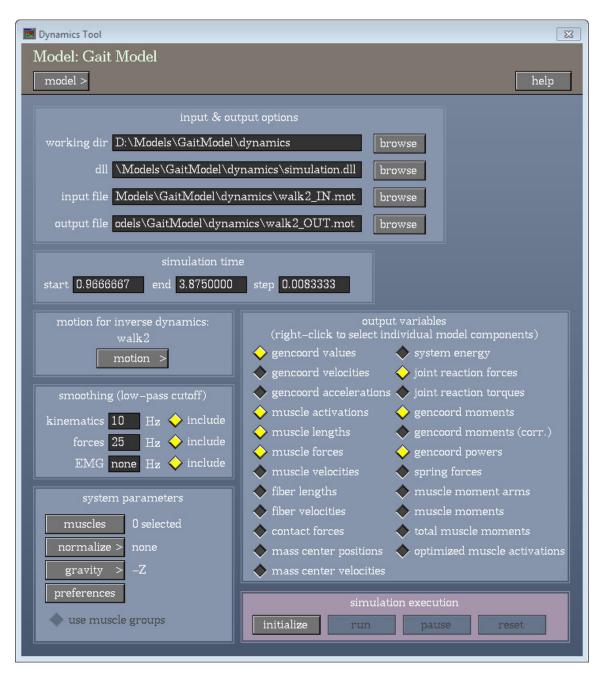


Figure 2-23. Dynamics Tool window

These pre-built DLLs include models of a full body (with simplified shoulder motion and lumped-mass hands), a lower-extremity model with lumped-mass upper body, and a right-arm model (with lumped-mass hand).

When a DLL is created, the following model properties are compiled into it, and thus you cannot change any of them in your SIMM model without creating a new DLL:

- the number of body segments
- the number of joints
- the number of degrees of freedom in any joint
- the order of rotations in a joint.

The following model properties are passed from SIMM to the DLL when the **initialize** button is pressed, and thus you can change them in SIMM without needing to reload or recreate a DLL:

- the number of muscles
- any property of any muscle
- the mass, mass center, or inertia of any body segment
- the clamped and locked states, and range of motion, of any gencoord
- the restraint moments at any gencoord
- the orientations of any joint's rotation axes
- contact objects (rigid body, spring-based, and NURBS-spring)
- force mattes (for applying external forces to the body segments)

2.16.1 Selector Menu

model

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the **help** button opens a window containing helpful text about the Dynamics Tool. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.16.2 Input and Output Options

working dir

This specifies the directory to be used for the dynamic simulation. This directory is used as the default location for the input and output files. For inverse dynamics, you should set the working directory before choosing the motion so that the paths for the input and output files will be set accordingly (see Section 2.16.4, Motion For Inverse Dynamics, for more details).

DLL This field specifies the DLL (dynamic link library) to be used for the dynamic simulation. This DLL can either be one that you create using the Dynamics Pipeline, or a prebuilt one that comes with the FIT Module. Make sure to select the one that matches the model you have loaded into SIMM.

input file

This field specifies the input file to be used for the dynamic simulation. For forward dynamics, this file is optional. If used, it can contain external forces and muscle excitation patterns that you want to apply during the simulation. For most inverse dynamics simulations, you do not need to specify an input file. When you select a motion (see Section 2.16.4, Motion for Inverse Dynamics), SIMM will automatically fill in the *input file* field with the name of a new file.

SIMM will create this file with all the appropriate data columns and smoothing parameters when you initialize the simulation.

output file

This field specifies the output file created by the dynamic simulation. As the simulation proceeds, it will write each time frame of output data to this file. When you select a motion for inverse dynamics (see Section 2.16.4, Motion for Inverse Dynamics), SIMM will automatically fill in the *output file* field with the name of a file based on the name of the input motion.

2.16.3 Simulation Time

The fields in this region control the time of the dynamic simulation, and how often the results are written to the output motion. For most inverse dynamics applications you do not need to set these parameters; they will be set automatically when you choose a motion (see Section 2.16.4, Motion for Inverse Dynamics).

start

This specifies the starting time for the dynamic simulation. For forward dynamics, set this field to the desired starting time (usually 0.0). When you choose a motion for inverse dynamics, the starting time will automatically be filled in with the time of the first frame of data, and you will not be allowed to change the value (you must crop the motion to change the start time).

end

This specifies the ending time for the dynamic simulation. For forward dynamics, set this field to the desired ending time. When you choose a motion for inverse dynamics, the ending time will automatically be filled in with the time of the last frame of data, and you will not be allowed to change the value (you must crop the motion to change the end time).

step

This specifies the step size, or reporting interval, for the dynamic simulation. As the simulation proceeds, it will write data to the output file, and return the same data to SIMM, every *step* seconds. When you choose a motion for inverse dynamics, the step size will automatically be filled in with the step size of the data in the motion, and you will not be allowed to change it.

2.16.4 Motion for Inverse Dynamics

motion

This button lets you choose a motion to use for the dynamic simulation (primarily for inverse dynamics). When you press this button, a pop-up menu is displayed with an option called *no motion*, followed by the list of motions currently loaded for this model. For forward dynamics, you will usually want to choose the *no motion* option. However, if you would like to prescribe the motion of some of the gencoords during a forward simulation, then choose the appropriate "partial" motion. For inverse dynamics, choose a motion with data for all of the gencoords. When you choose a motion, SIMM will automatically fill in the input file field with the name of a motion file that SIMM will generate. This file will contain the same data as the chosen motion, but will also include the smoothing parameters you have chosen (see Section 2.16.5, Smoothing (Low-Pass Cutoff)). SIMM will generate this file when you press the **initialize** button. When you choose a motion, SIMM will also fill in the *output file* field with the name of a file in the working directory that will hold the results of the simulation. Lastly, when you choose a motion SIMM will fill in the three simulation time fields with time values taken from the motion data

2.16.5 Smoothing (Low-Pass Cutoff)

The following smoothing parameters are used only for smoothing data that is contained in the motion chosen with the **motion** button. If you do not choose a motion, but instead specify an input file that you created for the simulation, you can perform smoothing of that data only by entering the appropriate column labels in the file itself.

kinematics

This specifies the low-pass cutoff frequency for smoothing the gencoord data. Enter a value of -1 for no smoothing.

forces

This specifies the low-pass cutoff frequency for smoothing the external force data (e.g., ground-reaction forces). Enter a value of -1 for no smoothing.

EMG

This specifies the low-pass cutoff frequency for smoothing the EMG data. Enter a value of -1 for no smoothing.

Use the **include** buttons to specify whether or not to include the various columns of data (e.g. if kinematics are not included, any gencoord information in the input file will be ignored).

2.16.6 System Parameters

muscles >

This button lets you choose which muscles you want to include in the dynamic simulation. If the *use muscle groups* box is checked and there are optimization groups defined for the muscles (see below), each muscle name will be colored according to the groups it is in.

normalize

This button lets you choose the normalization option for the output motion created by the dynamic simulation. Normalization of the output can be useful if you want to compare several dynamic analyses to each other. By default, normalization is off, and the output motion will contain the appro-

priate number of data frames as specified by the *simulation* time parameters. The option 0.0 -> 0.99 will result in the output motion containing 100 frames of data. In addition, a "reference" number will be added to each frame of data, ranging from 0.0 to 0.99. The option 0.0 -> 99.0 will also result in 100 frames, but the reference number will range from 0.0 to 99.0. Normalization does not affect the accuracy or step size of the numerical integration during the simulation. It only affects the time interval between the frames of data stored in the output motion.

gravity >

This button lets you specify the direction of the gravity vector. In general you should specify the gravity vector in your joint file, in which case that value is used to initialize the parameter in this tool. However, if you want to change it from the default, you can do so with this button. If you select the *custom* option, a dialog box will pop up allowing you to specify the x, y, and z coordinates of the gravity direction vector.

preferences

This button will display a dialog box with a list of the dynamic preferences that were defined in the joint file. These can be edited using either the checkbox to turn preferences on or off or editing the values in the fields provided.

use muscle groups

This checkbox lets you choose whether or not the muscles included in the simulation should be grouped during static optimization. If selected, all the muscles in each optimization group will be assigned the same activation level. An optimization group is a muscle group whose name ends in "_opt". When *use muscle groups* is selected, all of the muscle selection dialog boxes in the Dynamics Tool show the muscle names in a separate color for each group, and with the group number after the name. If there are muscles in the simulation that are not in an optimization group, they will be optimized individually.

2.16.7 Simulation Execution

initialize

This button initializes the dynamic simulation. A copy of the current model is sent to the simulation, which checks to make sure it matches the one built into the DLL. If the number and connectivity of the body segments, as well as the joint types, match between the two models, the simulation can proceed. The simulation model is updated with the mass properties and joint kinematics from SIMM, and an output motion is created to hold the data calculated by the simulation. This motion is added to the SIMM model, and a slider for it is added to the Model Viewer. As well, the files containing information about rigid body and NURBS contact, dllcontactparams.txt and dllnurbsparams.txt are written out.

run

This button starts the dynamic simulation. As it proceeds, calculating each time step of data, SIMM stores the frames of data in the motion that was created during the *initialize* step (see above). SIMM also updates the display of the model and the plots with the new data. When the simulation has completed, you can review the results using the motion's slider in the Model Viewer, and by plotting motion curves with the Plot Maker.

pause / resume

This button pauses a running simulation, or resumes a paused simulation. When paused, your only options for controlling the simulation are to resume it, or reset it (which ends the simulation, saving the partial results in the output motion).

reset

This button ends a simulation, and resets the dynamics to the uninitialized state. If the simulation has been initialized but is not yet running when the **reset** button is pressed, the motion created to hold the results is deleted from the model. If the simulation is running when **reset** is pressed, the results calculated so far are stored in the motion.

2.16.8 Output Variables

The checkboxes in this region specify which variables are calculated during the simulation, and saved in the output motion. When the **initialize** button is pressed, the output motion is created with the appropriate columns to hold all of the selected variables. These columns are initially filled in with zeros, and are overwritten with the simulation results as each time frame is calculated. To select or deselect an output option, click the left mouse button on the checkbox. To select individual model components for each output option, click the right mouse button on the checkbox and choose components from the dialog box. For example, if you right-click on gencoord values SIMM will display a dialog box allowing you to select which gencoord values you want to output. If you right-click on muscle lengths SIMM will display a dialog box of the muscles in the model. Below is a list of the available output variables.

gencoord values

the values of the generalized coordinates. In most cases you will want to keep this option on so that you can see the results of the simulation. However, one case if which you would not is when you are outputting *gencoord moments* (corr.). See the description of that option for details.

gencoord velocities

the velocities of the generalized coordinates.

gencoord accelerations

the accelerations of the generalized coordinates.

muscle activations

the dynamic activation levels of the muscles. This will not be the same as the excitation levels if the muscle model used in the dynamic simulation models non-linear activation dynamics (most models do).

muscle lengths

the musculotendon lengths of the muscles in the simulation.

muscle forces

the dynamic forces in the muscles in the simulation.

muscle velocities the velocities of the muscles in the simulation.

fiber lengths the fiber lengths of the muscles in the simulation.

fiber velocities the fiber velocities of the muscles in the simulation.

contact forces the forces applied by the rigid-body contact detection algorithm and/or the NURBS-spring contact detection algorithm. This includes the impulses applied during penetrating contact, and the forces applied during resting contact.

mass center positions the X, Y, Z coordinates of the location of each segment's mass center. This is usually needed only when you create a new SIMM model and want to double check that it has been

assembled correctly in the dynamics code.

mass center velocities the X, Y, Z components of the velocity of each segment's

mass center. This is usually needed only when you create a new SIMM model and want to double check that it has been

assembled correctly in the dynamics code.

system energy the total energy (potential plus kinetic) of the model during

the simulation. If there are no muscle, external, or friction forces applied, system energy will be conserved during the

simulation.

joint reaction forces the reaction forces at the joints. These forces are those

applied by the proximal segment to the distal segment. They are applied at the origin of the distal segment, and are

expressed in the distal segment's reference frame.

joint reaction torques the joint reaction torques at the joints. These are the torques

applied by the proximal segment to the distal segment, expressed in the distal segment's reference frame. They include user-applied torques such as joint restraint torques.

gencoord moments the moments applied to the gencoords by the simulation to

the moments applied to the gencoords by the simulation to generate the prescribed motion. These moments are calcu-

lated after all system forces (*e.g.*, gravity) and user-applied forces (*e.g.*, muscle, ground-reaction forces) are applied. In a typical inverse dynamics analysis, the gencoord motion is fully prescribed and no muscles are included. The gencoord moments are thus equal to the net moments required (*e.g.*, by the subject's muscles) to produce the prescribed motion. If you include muscles in your inverse dynamics analysis, your gencoord moments will be different. If you include exactly the right set of muscles and excitations (as was used by the subject in the recorded motion), no additional moments will be needed to produce the prescribed motion, and the gencoord moments will be zero. In a simulation that has no prescribed motion (*e.g.*, forward dynamics), the gencoord moments will be zero.

gencoord moments (corr.)

the moments required to generate the input motion, as calculated by a PD controller. Instead of prescribing the inverse dynamics motion using constraints (as the *gencoord moments* option does), the PD controller tracks the input motion by using the errors between the current and desired kinematics to apply corrective moments to the gencoords. This method depends on the specification of appropriate PD stiffness values for each gencoord. See Section 5.3 of the Dynamics Pipeline manual for a complete description of the PD controller method.

gencoord powers

the powers of the generalized coordinates. Power is defined as the moment multiplied by the velocity.

spring forces

the forces applied by the spring/floor objects. Each force is expressed in the reference frame of the corresponding spring point's body segment. There is no right-click option for this output since the number of spring forces output is specified in the simulation code.

muscle moment arms

the moment arms of the muscles in the simulation. The right-click dialog allows you to select which muscles you

want to output. The simulation will output the moment arms of the selected muscles for every generalized coordinate.

muscle moments

the moment that each muscle applies to the gencoords. The right-click dialog allows you to select which muscles you want to output. The simulation will output the moments of the selected muscles for every generalized coordinate.

total muscle moments

the total (net) moment applied to the gencoords by all of the muscles. The right-click dialog allows you to select which gencoords you want to output. The simulation will output the total muscle moment of all muscles for each selected gencoord.

optimized muscle activations

the dynamic activations of the muscles, as calculated by a static optimization algorithm. This algorithm estimates the muscle activations required to generate the input motion. At each time step in the simulation, the inverse dynamics gencoord moments are first calculated. Then a least-squares solver is used to determine the activations that will result in the muscles exerting the calculated moments. The optimization is static in the sense that it is performed independently at each time step, without regard to activation dynamics, but it does take into account the velocities of the muscle fibers. See Section 5.5 of the Dynamics Pipeline manual for a detailed description of the algorithm. The right-click dialog allows you to select which gencoords and muscles you want to include in the optimization. The simulation will optimize the activations of all of the specified muscles in order to generate the inverse dynamics moments calculated for the specified gencoords. Muscles that are not optimized but still included in the simulation will behave like "regular" muscles and will apply forces to the body segments according to their specified activation patterns (if any) and force model. If you want to output the activations of muscles that are not being optimized, you must select them for output using the muscle activations output variable.

2.17 Motion Editor

The Motion Editor (Figure 2-24) allows you to modify the motions linked to each model. The currently selected motion can be renamed, deleted, cropped, written to a file, or have events added to it. If plots have been created using these motions before they were edited, these existing plots will not change to reflect the new motion characteristics; the changes will only appear in newly created plots. This tool can be useful if you want to compare several motion trials to each other because it allows you to normalize them according to certain events, such as heel strike and toe off.

2.17.1 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Motion Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.17.2 Command Menu

motion

When you select this item a pop-up menu is displayed of motions linked to the model. Selecting a motion makes it the current motion for editing. Its properties can then be viewed and edited using the forms and sliders in the tool window.

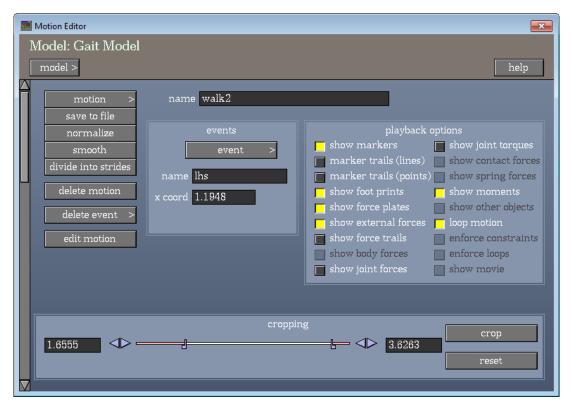


Figure 2-24. Motion Editor window

save to file

This command allows you to save the current motion to a file. When you select it, a file browser will let you choose the output file to create or overwrite.

normalize

This command normalizes the current motion so that it contains 101 time frames. It fits a natural cubic spline to each column of data, and evaluates it at 101 evenly-spaced intervals. It also adds a new column called "percent" to the motion, with values from 0.0 to 100.0. This column becomes the new X column, meaning that it will be used at the X axis when plotting data from the motion.

smooth

This command smooths the data in the current motion. It fits a GCV spline to each column of data, and evaluates it at the existing time frames. The low-pass cut-off frequency used for the smoothing has a default value of 6.0, but can be specified with the MOTION_SMOOTHING_FREQUENCY preference.

divide into strides

This command divides the motion into a set of smaller motions, one for each complete left and right gait cycle. A gait cycle is defined as the motion frames between two heelstrike events of the same foot. The original motion is preserved, and a new motion is created for each gait cycle. If there are no heelstrike events in the motion, this command does nothing.

delete motion

This command deletes the current motion. The motion is unlinked from the model and it will no longer be able to be viewed or plotted. Any existing plots using the deleted motion will remain unchanged.

delete event

This command lets you delete the events in the current motion. From the pop-up menu you can choose any one event, or *delete all* at the bottom to delete all of the events.

edit motion

This command allows you to edit the columns of data in the current motion. When you select it SIMM displays a dialog box showing all of the columns. If you select one and click *OK*, SIMM fits a GCV spline to the data for that column and displays it in the Function Editor.

If the motion contains kinematic data for a closed-loop model, it is recommended that you not edit data for gencoords in the loop. Doing so will violate the loop constraint and the model will not move properly (if IK is turned on, SIMM will adjust all of the gencoords to satisfy the loop constraint).

2.17.3 Cropping a Motion

You can change the range of an existing motion using this feature. *Start* and *end* frames can be entered into the number fields, or you can use the slider to move the *start* and *end* frames up and down until you have chosen the range you would like (the model will move in the motion window accordingly). Use the **crop** button to crop the motion when you are satisfied with the new range.

crop

This command crops the motion between the two selected frames. This operation cannot be undone. Any parameter fields in other tools that are linked to the cropped motion will be updated accordingly (*e.g.*, the *x min* and *x max* fields in the Plot Maker). Any existing plots of the motion will remain unchanged.

reset

Before you have cropped the motion you can use this command to reset the crop limits back to the full range of the motion.

2.17.4 Motion Events

You can create and edit events in a motion using this feature. You can edit the event's name and the X coordinate (e.g., time, percent of gait cycle) at which it occurs in the motion. New events created using this feature will not appear in any existing plots, but will appear in any new plots created from the motion.

event

When you select this item a pop-up menu of existing events is displayed. The first item in the pop-up menu, *new event*, creates a new event named *event_X*, adds it to the current motion at time 0.0, and makes it the current event for editing.

name

This field contains the name of the event. Events that SIMM calculates are labeled 'r' or 'l' for right and left, followed by 'hs' for heel strike and 'to' for toe off.

x coord

This field contains the value of the X coordinate at which the event occurs. For motions that have been normalized, the X coordinate is "percent of gait cycle". For most other motions the X coordinate is time, in seconds.

2.17.5 Playback Options

The checkboxes on the right side of the Motion Editor window allow you to change some display and evaluation options when animating a motion. Each option is described below.

show markers

This option toggles the display of marker data in the motion, which are usually shown as blue spheres. This option does not affect the display of the markers attached to the model (the pink spheres). If there is no marker data in the motion, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the markers in the model. The ones that have data in the motion are checked. You can use this dialog to select which markers you want to display during playback.

show marker trails (lines)

This option toggles the display of marker trail lines, which are white lines showing the trajectory of each marker during the entire motion. If there is no marker data in the motion, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the markers in the model. The ones that have data in the motion are checked. You can use this dialog to select which marker trails you want to display during playback.

show marker trails (points)

This option toggles the display of marker trail points, which are light-blue spheres showing the positions of each marker at every time frame during the motion. If there is no marker data in the motion, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the markers in the model. The ones that have data in the motion are checked. You can use this dialog to select which marker trails you want to display during playback.

show foot prints

This option toggles the display of foot prints, which are displayed on the floor to represent the position of the foot during stance. Each foot print is calculated as follows. First, the position of the heel marker at heel strike and the position of the toe marker at toe off are calculated in the ground reference frame, and the Z values are set to 0.0 so that the points are on the floor. A foot-print motion object is then scaled, translated, and rotated so that the back of its heel is at the heel marker position, and the base of its second toe is at the toe marker position. With this method, the size of each foot print during a motion will be approximately equal to the size of the subject's foot, but the foot prints in a motion may not all be the same size. In fact, if the subject slides or drags his/her foot on the floor after heelstrike and before toe off, the foot print will be much larger than normal. Thus the size of the foot prints, in addition to their locations, can be used to help analyze the gait pattern.

show force plates

This option toggles the display of force plates, which are usually shown as blue rectangles on the floor. Force plates are defined in the motion file header, and are added there when a motion capture trial (TRB/TRC/C3D file) is imported into SIMM. If there are no force plates defined in the header, this option is grayed out.

show external forces

This option toggles the display of external forces in the motion data. These forces are usually ground-reaction forces, but also include any force expressed in the ground frame. If there are no external forces in the motion data, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the external forces in the motion, which you can use to turn on or off the display of individual forces.

show force trails

This option toggles the display of force trails, which show the trajectories of the external forces during the motion. A force trail consists of a white line representing the force at each time frame, and a series of line segments connecting the tips of the white force vectors. If there are no external forces in the motion data, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the external forces in the motion, which you can use to turn on or off the display of individual force trails.

show body forces

This option toggles the display of body forces in the motion data. Body forces are forces that are applied to a specific body segment in the model. If there are no body forces in the motion data, this option is grayed out. You can right-click on the checkbox to display a dialog box showing all of the body forces in the motion, which you can use to turn on or off the display of individual forces.

show joint forces

This option toggles the display of joint reaction forces in the motion data (if there are none, this option is grayed out). You can right-click on the checkbox to display a dialog box showing all of the joint reaction forces in the motion, which you can use to turn on or off the display of individual forces. The dialog also contains a checkbox called *Display as arrows*. This controls whether the joint reaction forces are displayed as arrows or as translucent spheres. When displayed as arrows, you can select which joints you want to display. For each joint selected, the XYZ components of the reaction force are shown as a single arrow. When the forces are displayed as spheres, each force component (X, Y, and Z) is displayed as its own sphere, with red for X, green for

Y, and blue for Z. Each sphere is centered on the joint center, and drawn partially transparent to help see overlapping spheres.

show joint torques

This option toggles the display of joint reaction torques in the motion data (if there are none, this option is grayed out). You can right-click on the checkbox to display a dialog box showing all of the joint reaction forces in the motion, which you can use to turn on or off the display of individual forces. The dialog also contains a checkbox called Display as arrows. This controls whether the joint reaction torques are displayed as arrows or as translucent spheres. When displayed as arrows, you can select which joints you want to display. For each joint selected, the XYZ components of the reaction torque are shown as a single arrow. When the torques are displayed as spheres, each torque component (X, Y, and Z) is displayed as its own sphere, with red for X, green for Y, and blue for Z. Each sphere is centered on the joint center, and drawn partially transparent to help see overlapping spheres.

show contact forces

This option toggles the display of contact forces in the motion data (if there are none, this option is grayed out). Contact forces are the forces calculated by the rigid-body contact algorithm and by the NURBS-based contact algorithm in the Dynamics Pipeline. You can right-click on the checkbox to display a dialog box showing all of the contact forces in the motion, which you can use to turn on or off the display of individual forces.

show spring forces

This option toggles the display of spring forces in the motion data (if there are none, this option is grayed out). Spring forces are the forces calculated by the spring-based contact algorithm in the Dynamics Pipeline. You can right-click on the checkbox to display a dialog box showing all of the spring forces in the motion, which you can use to turn on or off the display of individual forces.

show moments

This option toggles the display of moments that are applied to the generalized coordinates. These moments are usually the results of inverse dynamics calculations, but you can also display moments that you apply to the generalized coordinates during a forward simulation. The dialog also contains a checkbox called *Display as arrows*. This controls whether the gencoord moments are displayed as arrows or as translucent spheres. When displayed as arrows, the arrow for each gencoord is drawn along the gencoord's axis, with the length representing the magnitude of the moment. When displayed as spheres, a red sphere is displayed for each moment, centered on the origin of the child segment of the gencoord's primary joint.

show other objects

This option toggles the display of all motion objects that are not covered by the above options, such as the spheres used to represent mass center positions in motion files created by dynamic simulations.

loop motion

This option toggles the looping of the motion playback. If turned on, the motion will wrap around to the beginning when it gets to the end. The motion will thus loop continuously until you stop it.

enforce constraints

This option controls whether or not to enforce constraint objects while animating the motion. If the motion contains complete gencoord data that already satisfy the constraint objects, then you can turn off this option so that SIMM does not recalculate them. If your model does not contain any constraint objects, this option is grayed out.

enforce loops

This option controls whether or not loop constraints are enforced while animating the motion. If the motion contains complete gencoord data that already satisfy the loop constraints, then you can turn off this option so that SIMM does not recalculate them. If your model does not contain any closed loops of body segments, this option is grayed out.

show movie

This option toggles the display of the movie [if any] associated with the motion. See Section 3.9, Movie Files, for information on how to load movies into SIMM and link them to motions.

2.18 Motion Reporter

The Motion Reporter (Figure 2-25) allows you to create reports from sets of motions. A report is an Excel file containing means and standard deviations of various variables calculated from the motions. This tool can create reports of any kind of motion (*e.g.*, walking, running, golf swing), but has some additional features designed specifically for creating gait reports.

The report file that the Motion Reporter creates is a binary Excel file with the data organized into sheets (e.g., one for angles, one for moments, one for muscle lengths). You can post-process or format the data any way you want by extracting it from this file. However, if you create reports using the Gait Model that comes with SIMM (in the *Mod*els/GaitModel folder), there is a corresponding Excel graphs file containing formatted graphs that look like a standard gait report. This file is called GaitModel report.xls. It links to a file called *GaitModel data.xls* to get the subjectspecific report data, and it links to NormalData.xls to get data for normal gait, which it also displays in the graphs. Thus, if you use the Motion Reporter to create a file called GaitModel data.xls and put it in the same folder as GaitModel report.xls, you can then open GaitModel report.xls in Excel, select the option to "update" links," and see the report data in the updated graphs.

If you choose to include kinetic variables in a report (moments, powers, joint forces), the Motion Reporter works

in conjunction with the Dynamics Tool to perform dynamic simulations that calculate those variables. Thus, to report kinetic variables you must have the Dynamics Pipeline Module or the FIT Module, and have already set up the Dynamics Tool with the appropriate parameters.

If the Motion Reporter needs to perform a dynamic simulation on a motion, it will do so using the current settings in the Dynamics Tool (except for the *motion for inverse dynamics*, which it will set automatically). Thus, the work-

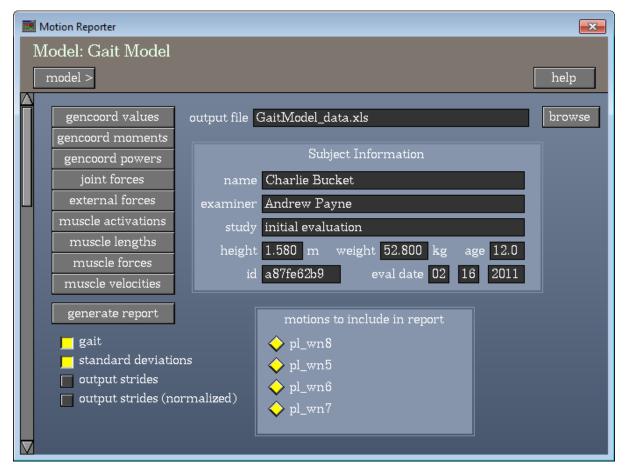


Figure 2-25. Motion Reporter window

ing directory, the DLL, the output variables, the system parameters, and the smoothing parameters must all be set appropriately before creating the report.

Because the dynamic simulations are performed through the Dynamics Tool, you have an opportunity to customize them. For example, in a standard gait report you might include graphs of EMG activity for the few muscles that were recorded. With the Motion Reporter and Dynamics Tool, you can choose to include all of the muscles in the simulation, and optimize their activations [to generate the inverse dynamics moments]. The gait report will then contain graphs of estimated activations for all of the muscles (with some guided by the input EMG).

If the *gait* option is turned off, SIMM will not perform any cropping or vetting of the motions before creating the report. It will assume that each selected motion is of the same activity, with approximately the same starting and ending positions. It will perform a dynamic simulation on each motion (if needed) and then normalize it to contain 101 time steps. It will then calculate means and standard deviations of all of the selected report variables.

If the *gait* option is turned on, SIMM will assume that each motion represents one or more walking strides, indicated by heel strike and toe off events. If dynamic simulations are required, it also verifies that each stance phase of a stride occurs on a force plate, so that correct inverse dynamics moments can be calculated. If a part of stance occurs off of a force plate, that stride is not included in the report.

2.18.1 Generating a Report

When you generate a motion report, SIMM performs the following steps:

- (1) Examine each selected motion to see if it already contains the kinetic variables (*e.g.*, moments, powers) chosen for output to the report. If it is missing any of them, the motion is flagged as requiring a dynamic simulation.
- (2) If any dynamic simulations are required, set up the simulation module using the current settings in the Dynamics Tool. The output variables should be set so that they include all of the variables needed for the report, and the set of muscles to include in the simulation should be set appropriately. For example, if you want only muscle lengths in the motion report, then you do not need to include muscles in the simulation. However, if you want optimized muscle activations in the report, then you need to select the *optimized muscle activations* output variable, and choose the appropriate set of muscles.
- (3) Run a dynamic simulation on each motion that requires it. If the *gait* option is not selected, the simulation is performed on the entire motion. In this case you are responsible for making sure that the motion is cropped appropriately (*i.e.*, that there are no time frames involving unrecorded external forces). If *gait* is on, SIMM divides each motion into left and right strides based on heel strike and toe off events. The original motion is then replaced by the individual stride motions. It then checks each stride to see if the entire stance portion occurs on a force plate. If it does not, the stride is excluded from the dynamic simulation. If it does, the simulation is performed and the output variables are added to the motion.
- (4) If *gait* is on, calculate stride and step information for each motion (*e.g.*, stride length, step width), as well as foot progression angle.

- (5) Normalize each motion so that it contains 101 time frames. Each column of data in the motion is interpolated with a cubic spline, and sampled at 101 evenly-spaced time steps. It also adds a new column called "percent" to the motion, with values from 0.0 to 100.0.
- (6) Check each motion to see if it contains all of the selected muscle output variables (activations, lengths, forces). Calculate the missing ones using SIMM's isometric muscle model and add the results to the motion.
- (7) Calculate the mean (and standard deviation, if requested) of each selected output variable. Write this information to a binary Excel file, organized with sheets for each type of output.

2.18.2 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Motion Reporter. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

2.18.3 Command Menu

gencoord values

This command lets you select which gencoord values you want to include in the report. Clicking on the button displays a dialog box containing all of the gencoords in the model.

gencoord moments

This command lets you select which gencoord moments you want to include in the report. Clicking on the button displays a dialog box containing all of the gencoords in the model.

gencoord powers

This command lets you select which gencoord powers you want to include in the report. Clicking on the button displays a dialog box containing all of the gencoords in the model.

joint forces

This command lets you select which joint reaction forces you want to include in the report. Clicking on the button displays a dialog box containing all of the joints in the model.

external forces

This command lets you select which external forces (usually ground-reaction forces) you want to include in the report. Clicking on the button displays a dialog box containing all of the force mattes in the model (external forces expressed in the ground frame are applied to the body segments using force mattes).

muscle activations

This command lets you select which muscle activations you want to include in the report. The output activation for each muscle is the average of its activation for each motion trial. The activation for a particular motion trial depends on several factors. If there is no EMG data for a muscle, its activation is whatever it is set to in the Muscle Editor (default = 1.0). If there is EMG data but the muscle is not included in the dynamic simulation of that motion trial, then the activation is the smoothed EMG signal. If the muscle is included in the simulation, its activation is calculated using the input excitation and the equations for activation dynamics. The input excitation is the EMG signal (if any), the specified excitation pattern (if any), or whatever the activation level specified in the Muscle Editor. Lastly, if the dynamic simulation calculates optimized muscle activations, then the acti-

vations written to the motion report will be the optimized ones.

muscle lengths

This command lets you select which muscle lengths you want to include in the report. Muscle lengths are calculated from the kinematic data (*i.e.*, gencoord values), so you can output them without including the muscles in the dynamic simulation of the motion trial.

muscle forces

This command lets you select which muscle forces you want to include in the report. The output force for each muscle is the average of its force for each motion trial. The force for a particular motion trial depends on whether the muscle is included in the dynamic simulation of that trial. If it is included, then the force is the dynamic force calculated by the simulation. If it is not included, then its force is the isometric force that SIMM calculates based on its length and activation. See the description of the *muscle activations* command for information about how the activation of a muscle is calculated.

muscle velocities

This command lets you select which muscle velocities you want to include in the report. Muscle velocities are calculated from the kinematic data (*i.e.*, gencoord values), so you can output them without including the muscles in the dynamic simulation of the motion trial.

generate report

This command generates the motion report, using the currently selected motions and output variables. See Section 2.18.1 Generating a Motion Report, for information on how the report is generated.

2.18.4 Forms

Output File Form

output file

The name of the report file to generate. The generated file will be a binary Excel file with the data organized into sheets. If you are using the Gait Model that comes with SIMM, you should name this file <code>GaitModel_data.xls</code>. The Gait Model includes an Excel file of formatted graphs called <code>GaitModel_report.xls</code>. This graphs file links to <code>GaitModel_data.xls</code> to get the data for the graphs (as well as <code>NormalData.xls</code> for normal data). Thus if you generate a report called <code>GaitModel_data.xls</code>, you can then open <code>GaitModel_report.xls</code> in Excel, select the option to "update links," and see the report data in the updated graphs.

Subject Information Form

name The name of the subject or patient.

examiner The name of the examiner, technician, supervising clinician, physical therapist, etc.

study A brief description of the study or condition.

height The height of the subject. This value is not used for any model calculations; it is merely copied to the motion report.

weight The weight of the subject. This value is not used for any model calculations; it is merely copied to the motion report.

age The age of the subject.

id The subject ID. This can be any string of characters or numbers.

test date The date the motion trials were collected. The format day/ month/year is preferred, but since this information is not used by SIMM and merely copied to the motion report, *month/day/year* is also acceptable.

2.18.5 Toggle Buttons

gait

This toggle button controls whether or not SIMM should treat the motions in the report as gait strides. If this button is turned on, SIMM will divide each motion into left and right strides based on heel strike and toe off events. It will then process and average the left and right strides separately, and calculate all left-side output variables from the left strides, and all right-side output variables from the right strides. Right-side variables are denoted by the suffix " r" (e.g., knee flex r, semimem r, hip r), and left-side variables are denoted by the suffix " 1". Thus, this gait option must be used in conjunction with a model that has appropriately defined components. Additionally, if SIMM performs simulations on the strides to calculate dynamic variables such as moments and powers, it will check to make sure that the stance portion of the stride occurs on a force plate. If it does not, that stride will be excluded from the motion report.

If the *gait* button is turned off, SIMM will not crop or divide the motions based on events. Also, if it performs simulations on the motions to calculate dynamic variables, it will not check that all ground contact occurs on force plates. It will process each motion trial without cropping, normalize it to 101 time frames, and then compute averages and standard deviations.

standard deviations

This toggle button lets you choose whether or not to include standard deviations in the motion report. If it is turned on, each column of output variables (the means) in the report file will be followed by a column of data representing one standard deviation from the mean. output strides

This toggle button is active when the *gait* button is on. It controls the output of motion files containing the individual strides that are found during report generation. A motion file is written for each stride, with a name constructed from the motion name and the stride identifier (e.g., "r2" for second right stride). The files are written to the folder containg the gait report.

output strides (normalized)

This toggle button is active when the *gait* button is on. It works the same way as the *output strides* button described above, except that each stride is normalized to contain 101 time steps.

2.19 Skin Editor

The Skin Editor allows you to create, edit, and export deformable skins. A skin is similar to a polyhedral bone in that it consists of a set of vertices and a set of [usually] planar polygons using the vertices. However, in a skin each vertex can translate relative to all of the body segments as the joints of the model are moved. The position of each skin vertex is calculated as the weighted sum of a set of attachment points that are rigidly attached to the body segments. For example, in a skin that covers a leg model, a skin vertex near the knee might have attachments on the thigh and the shank (and possibly the patella as well). As the knee flexes, each attachment moves with its body segment. The position of the skin vertex is calculated by determining the position of each attachment in the ground frame, mutiplying it by its weight, and averaging all of the weighted attachments together. With proper positioning of attachments and assigning of weights, the skin can be made to deform smoothly around the leg as the knee is flexed and extended.

A skin can be used to represent the external skin of an animal, but it can also be used to model muscle and ligament surfaces, clothing, or any other deformable surface. Skins are typically stored as Collada DAE files, but SIMM also supports a custom skin file format (SKN) that is similar to the Wavefront OBJ format. Skins can be added to any musculoskeletal model either by referencing them in JNT files, or by making them in the Skin Editor.

SIMM can make a skin by taking one of the bones in a model and determining, for each of its vertices, which body segment[s] the vertex is closest to. It then converts the bone vertex into a skin vertex, with attachments on the one or two closest body segments, and weights proportional to the distances to those segments. This process is a good first step in making a skin, but you will need to adjust the attachment weights manually to make the skin deform smoothly.

In addition to making skins, the Skin Editor allows you to modify them by selecting one or more vertices and changing their attachment locations and weights. You can then save your changes back to a DAE or SKN file.

2.19.1 Selector Menu

model >

The model selector lets you select the current model. If you never have more than one model at a time loaded into SIMM, then you do not need to use the model selector.

help

Clicking on the help button opens a window containing helpful text about the Skin Editor. When you are done reading it, close the window using its close box. You will be able to re-open it at a later time by selecting **help** again.

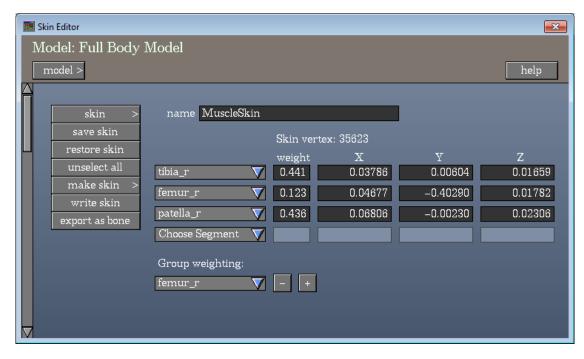


Figure 2-26. Skin Editor window

2.19.2 Command Menu

restore skin

When you select this item a pop-up menu is displayed of skins in the current model. Selecting a skin makes it the current skin for editing. Its properties can then be viewed and edited using the buttons and forms in the tool window.

This command saves the current state of the current skin in

This command saves the current state of the current skin in the model. It can be used in combination with the **restore** skin button to help you undo changes made while editing skins.

This command restores the most recently saved copy of the model's current skin. It can be used in combination with the **save skin** button to help you undo changes made while editing skins.

unselect all

This command unselects all of the currently selected skin vertices in the current skin

make skin

This command makes a skin from a static bone. To make a skin, select this command and choose a bone from the popup menu. SIMM will examine each vertex of the bone and calculate which body segments it is closest to. The distance between a vertex and a body segment is defined as the shortest distance between the vertex and all of the vertices on all of the bones of the body segment. If the vertex is more than three times closer to one body segment than all others, it is given only one attachment, with a weight of 1.0. Otherwise, the vertex is given two attachments, with weights proportional to the relative distances to the two body segments. All of the calculations are performed with the model in its current pose, so it is important to position the static bone and set all of the model's gencoord values appropriately before executing this command.

write skin

This command writes the current skin to a file. A file browser is displayed allowing you to select the type of file (DAE or SKN), as well as the name and location.

export as bone

This command creates a SIMM bone file containing the current skin in its current pose. All of the skin vertices are transformed into the ground frame, and then exported as a bone file. When you select this command, SIMM displays a file browser that lets you choose the name and type of the out file.

2.19.3 Skin Attachments

When one or more skin vertices are selected for editing, the Skin Editor displays GUI items for modifying the weights and XYZ locations of the attachments for those vertices. These combo boxes and number fields are located to the right of the command menu. Each row contains the properties of one attachment. If you select more than one skin vertex, the combo boxes and number fields will be blank if the vertex attachments do not have the same values. However, you can still enter a value to change a property for an attachment in all selected vertices.

segment

This combo box shows the name of the body segment for the attachment. You can change the segment by clicking on the name and choosing a different body segment from the pop-up menu. If more than one skin vertex is selected, and those vertices have attachments on different body segments (than the other selected vertices), then the combo box will show *Choose Segment* instead of a segment name. If you click on the combo box and choose a segment, that attachment for all of the selected skin vertices will be changed to the chosen segment.

weight

The weight of an attachment controls how closely the skin vertex follows it when the joints move. The sum of the weights for all attachments for each skin vertex must equal 1.0. If you click on the weight field for an attachment and enter a value, the weights of the other attachments, if any, will automatically be adjusted so that the sum is 1.0. If more than one skin vertex is selected, and those vertices have attachments with different weights (than the other selected vertices), the weight fields will be blank. If you click on a blank weight field and enter a value, the weight for that attachment for all selected vertices will be changed to that value.

X, Y, Z The XYZ coordinates of an attachment specify the location of the point in the reference frame of the body segment it is attached to. If more than one skin vertex is selected, and those vertices have attachments with different coordinates (than the other selected vertices), the X, Y, and/or Z fields will be blank. If you click on a blank field and enter a value, the coordinate for that attachment for all selected vertices will be changed to that value.

2.19.4 Group Weighting

The group weighting feature in the Skin Editor allows you to proportionally change attachment weights for a group of skin vertices, which can be helpful for smoothly deforming the skin around a joint. To use it, first select a group of skin vertices. Then select a body segment (the "target" segment) from the combo box located under the *Group weighting* label. Now when you press the + and - buttons next to the combo box, the attachment weights of the selected skin vertices are adjusted as follows. For each selected skin vertex, if the vertex has an attachment on the target segment, its weight is increased by 10% if the + button was pressed, and decreased by 10% if the - button was pressed. If the selected skin vertex does not have an attachment on the target segment, then if the - button was pressed the vertex's attachments are not changed. If the + button was pressed, a new attachment is added to the vertex. The location of the attachment is the current position of the skin vertex (expressed in the target segment frame), and the attachment weight is set to 0.05. In all cases, after the weights of the target segment attachments have been changed, the weights of all other attachments for the selected skin vertices are adjusted so that the weight total for each vertex is 1.0

2.19.5 Model-Window Commands



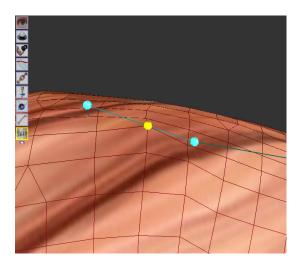
Skin vertices can be selected and edited using keyboard and mouse commands in the model window. To use these commands, you must first set the tool mode to the Skin Editor by clicking on the Skin Editor icon in the heads-up display (HUD) on the left side of the model window. To indicate that all keyboard and mouse commands are being directed to the Skin Editor, the icon will be highlighted with a yellow frame, and the title bar of the Skin Editor will be highlighted in yellow.

Note: you cannot select or unselect entire skins with model-window commands. You must use the **skin** button in the Skin Editor to select the current skin for editing.

SELECTING SKIN VERTICES

To select a skin vertex, press and hold the S key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Selecting skin vertices." As you move the cursor around the model window, if it hovers over a skin vertex, the vertex will be highlighted with a yellow wireframe sphere, and its index will be shown in a tool tip box. If you then left-click on the vertex, it will be added to the group of selected skin vertices. If you click on an already selected vertex, it will be unselected. You can also box-select skin vertices by pressing and holding down the left mouse button and dragging the cursor to form a rectangle around the vertices you want to select (or unselect).

As you select and unselect vertices, the Skin Editor window updates to show the number of vertices selected and the weights and locations of their attachments. If the selected vertices do not have the identical attachment segments, weights, or locations, the corresponding fields in the GUI are blank, indicating that multiple values are present.



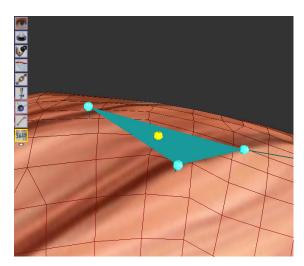


Figure 2-27. The left image shows a selected skin vertex (yellow sphere) on a line between its two attachments (green spheres, each fixed to a different body segment). The relative weights of the two attachments determine the location of the skin vertex on the line. In editing mode, the skin vertex can be dragged along the line to change the attachment weights. The right image shows a skin vertex with three attachments, which form a triangle. The location of the skin vertex is determined by the relative weights of the three attachments, and in eduting mode the vertex can be dragged anywhere inside the triangle to change the weights.

EDITING SKIN VERTICES



To edit the group of selected skin vertices, press and hold the E key. While this key is pressed, the HUD text box in the lower left corner of the model window displays "Moving skin vertices and attachments." Each selected vertex is displayed with a yellow sphere, and each of the vertex's attachments is displayed with a greenish blue sphere with a line to the origin of the body segment it is attached to. If the vertex has only one attachment, then the attachment sphere is coincident with the vertex sphere, and you will see only the latter. If the vertex has two attachments, a line is drawn between them, and the skin vertex is located somewhere along the line, depending on the weights. If the vertex has three attachments, a triangle is drawn between them, and the

skin vertex lies somewhere inside the triangle, depending on the weights. Note: if you cannot see the attachments for a skin vertex, they may either be coincident with each other and/or the vertex, or hidden underneath the skin itself. You can usually reveal them by changing a gencoord value to move the attachments apart. Figure 2-27 shows a typical view of a skin vertex with two attachments, and a vertex with three attachments.

To change the attachment weights of a skin vertex, put the cursor over the vertex (the yellow sphere) and press and hold the left mouse button. As you drag the vertex along the line (two attachments) or triangle (three attachments), the Skin Editor window will update to show the new weights. Release the left mouse button when you are done. Note: if the skin vertex has more than three attachments, you cannot interactively change the weights this way; you can only type in values in the Skin Editor window.

To change the location of a skin vertex attachment, put the cursor over the desired attachment (the greenish blue sphere) and press and hold the left mouse button. Drag the sphere to the desired location and release the left mouse button.

2.20 Function Editor

The Function Editor allows you to modify any function in a model, as well as create new functions. To edit a function, you must select it for editing in another SIMM tool, such as the Muscle Editor (for muscle functions) or the Joint Editor (for joint functions). The function is then displayed in the Function Editor window, allowing you to move, add, and delete control points. You can also change the type of a function; the supported types are: natural cubic, GCV

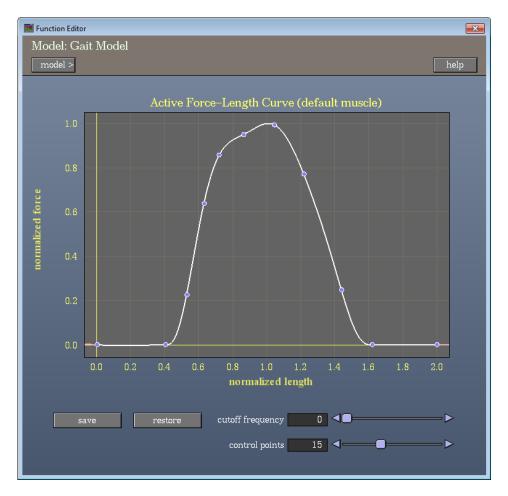


Figure 2-28. Function Editor window

spline, linear, and step. If a particular function in your model is being used by several model components (*e.g.*, the same restraint function might be used by multiple gencoords), when you modify it in the Function Editor it will modify the behavior of *all* model components that use it. If you want to modify a function only for the model component you selected it from, use the **clone** command, described below in the *Right-Click Menu* section. The Func-

tion Editor also allows you to switch the selected model component to use a different function or to create and use a new function. These commands are also described in the *Right-Click Menu* section.

2.20.1 Command Menu

save

This command saves a copy of the function being edited. It can be used in combination with the **restore** button to help you undo changes made while editing.

restore

This command restores the most recently saved copy of the function. It can be used in combination with the **save** button to help you undo changes made while editing.

cutoff frequency

This parameter is enabled when the function being edited is a GCV spline, which supports smoothing. The smoothing algorithm is a low-pass filter, and the parameter value is the cutoff frequency (all frequency components above this value are removed).

control points

This parameter is the number of control points the function has. If you type in a new value or move the slider to change it, the function will be resampled at regular intervals so that it has the specified number of control points.

This parameter is particularly useful when editing data columns from motions (see Section 2.17.2 for details). These columns typically contain hundreds of data points, making it tedious to modify a portion of the function. Instead of moving each of the original data points, you can use the *control points* parameter to undersample the function and then modify the reduced set of points. You do not need to restore the number of control points when finished; SIMM will automatically resample the modified function to calculate a data point for each time frame of the motion.

2.20.2 Function Editing

When a function is displayed in the graph area, you can add, delete, and move its control points. To move a control point, press and hold the left mouse button while the cursor is on the point. Drag the cursor to move the point to the desired location and then release the left mouse button. To select multiple points for moving, you can Shift-click on a point to add it to the set of selected ones, and Ctrl-click to toggle its selection. You can also box-select control points by pressing the left mouse button away from a point and dragging the cursor to form a box around the desired points. When you start to box-select, all currently selected control points will be deselected so that only the ones in the box will be selected. You can Shift-box-select to add the points in the box to the set of selected ones, and Ctrl-box-select to toggle the selection of points in the box. Once you have selected the desired set of control points, click on any one of them and drag them to a new location.

To help you edit a function, you can press the i, o, l, r, u, and d keys to move the view of the function in, out, left, right, up, and down, respectively. You can also use the x and y keys to zoom in and out along just the x or y axes (press the Shift key along with x or y to zoom out).

Depending on the function being edited, there may be enhancements to the display of the graph area to help with editing. For example, with gencoord restraint functions (Figure 2-29), the red regions mark the limits of the gencoord's range of motion (red lines if the gencoord is unclamped), and the yellow vertical line shows the current value of the gencoord.

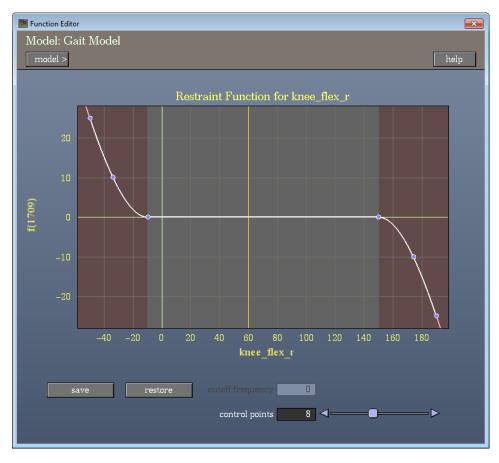


Figure 2-29. Gencoord Restraint Editing. This figure shows the Function Editor editing the restraint function for the knee_flex_r gencoord. The Y axis is Newton-meters of moment, and the X axis is degrees. The red regions on the ends indicate that the gencoord is clamped, so its value in SIMM cannot leave the gray region. During dynamic simulations, the gencoord can go outside this range, but when it does the restraint function applies a moment to push it back.

Note: If you create a function that has steep slopes or sharp corners, the display of the function may look choppy. That is, you may be able to notice the series of line segments that is used to approximate the curve in the graph area. This piece-wise linear approximation is for display purposes only. When SIMM needs to evaluate the function for use in the model, it finds an exact solution using the algebraic form of the function.

RIGHT-CLICK MENU

In addition to clicking on the function's control points and moving them, you can also modify the function by right-clicking in the editing area to reveal a pop-up menu with other editing commands. Here is a list of the available commands.

add point add knot point This command adds a control point to the function, at the location of the cursor when you right-clicked. If you right-click on an existing control point, the menu item is *add knot point*, and the point selected will be duplicated, creating a knot.

delete point

This item is enabled only if you right-click on a control point. It deletes the point, provided that there are enough remaining points to properly define the function (*e.g.*, two points for a linear function, six for a GCV spline).

change type

This command lets you change the type of the function that interpolates the control points. The available types are: *step function* (zero-order hold), *natural cubic spline* (3rd order), *GCV spline* (5th order), and [piece-wise] *linear*.

switch to

This command lets you specify which function in your model is used for this model component (the component you chose for editing in another tool). The submenu contains a list of all of the functions defined in your model (the one currently being used is indicated with a check mark). Functions that are defined in the muscle file have an 'm' before their numbers. When you switch to a different function, the previous one is not deleted, even if it is not being used by any model component.

clone

This command makes a copy of the current function and switches the model component to use the copy. This command is useful if a function is used by several model components but you want to modify it only for the one you selected for editing.

make new

This command creates a new function with two control points and switches the model component to use it. The previous function is not deleted, even if it is no longer used by any model component.

show/hide crosshairs

This item turns on or off the display of cross hairs when the cursor is in the function editing area. In addition to the cross hairs, the XY coordinates of their intersection are displayed as well.

reset view (control points)

This command resets the view of the function editing area so that all of the control points are visible in the window.

reset view (function)

This command resets the view of the function editing area so that all of the function is visible in the window.

enter XY coordinates

This item is enabled only if you right-click on a control point (or knot). It displays a dialog box with fields for entering X and Y coordinates. If you click on a knot, all of the control points in the knot will be moved to the new coordinates.

2.21 **Scripting Tool**

The Scripting Tool (Figure 2-28) lets you type SIMM commands into a console window, allowing you to perform simple batch processing. The functions in SIMM that are currently accessible via the Scripting Tool are all file open/ close commands, every function in the Model Viewer, Plot Maker, Motion Reporter, Dynamics Tool, and Motion Edi-

```
Scripting
                                                                                                ×
[pm]> load GaitAnalysis.txt
[pm]> cd "C:\Program Files\MusculoGraphics\SIMM50\GaitAnalysis"
[pm]> pref MOCAP_MODEL "C:\Program Files\MusculoGraphics\SIMM50\GaitAnalysis\dynamic.jnt"
[pm]> open_mocap init100.trb
[pm]> open walk1.trb
[pm]> tool mt
[mt]> set motion walk1
[mt]> divide steps
[mt]> set motion walk1_r1
[mt]> normalize
[mt]> set motion walk1_l1
[mt]> normalize
[mt]> tool pm
[pm]> open "C:\Program Files\MusculoGraphics\SIMM50\GaitAnalysis\NormalHamstrings.plt"
[pm]> yvar norm_musculotendon_length
[pm]> xvar walk1_r1
[pm]> semimem_r on
[pm]> semiten_r on
[pm]> make
[pm]> semimem_r off
[pm]> semiten_r off
[pm]> xvar walk1_l1
[pm]> semimem_l on
[pm]> semiten_l on
[pm]> make
[pm]>
```

Figure 2-29. Scripting Tool window

tor, and the save/restore functions in the Muscle Editor. These allow you to batch process motion trials, perform inverse dynamics analyses on them, and create plots of muscle lengths or other motion variables (*e.g.*, for gait analysis). More script functions will be available in future versions of SIMM.

The Scripting Tool uses a *current tool* and a *current folder* to help parse commands that you enter. When you enter a command, the first word in it is checked to see if it is a top-level command such as *open*, *save*, or *close*. If it is, the Scripting tool parses and executes the command. If it is not, the command is passed on to the current tool, such as the Motion Editor or Plot Maker. This tool then parses and exe-

cutes the command, printing an error message if there are any problems. When you enter a command with a filename that is not an absolute path starting with a drive letter, either for opening or saving a file, SIMM looks for the file relative to the current folder (e.g., open GaitModel.jnt). If you do enter an absolute path for a filename, the current folder is not used to look for the file (e.g., open C:\GaitAnaly-sis\Model\GaitModel.jnt).

2.21.1 Top-Level Commands

These commands are available in the Scripting Tool no matter which tool is the current one.

same as selecting *Open*... from the File menu. It displays a file browser so you can choose the file[s] to open.

opens the specified file[s]. The rules for opening multiple files at once are the same as if you selected them all in the file browser (e.g., you cannot select more than one muscle file at a time).

same as selecting *Open Mocap Model*... from the File menu; display a file browser so you can choose the static trial to open.

opens the specified static trial and displays the dialog box with import options.

closes (deletes) the specified model. SIMM does not prompt you to confirm the action.

closes (deletes) the specified plot. SIMM does not prompt you to confirm the action.

saves the specified model to the specified joints file. SIMM does not notify you if this action overwrites an existing file.

open

open <*file1*> <*file2*> ...

open mocap

open mocap <filename>

close <model>

close <*plot*>

save jnt <filename> <model>

save msl <filename> <model>

saves the specified model to the specified muscle file. SIMM does not notify you if this action overwrites an existing file.

save plt <filename> <plot>

saves the specified plot to the specified file. SIMM does not notify you if this action overwrites an existing file.

save smm <filename> <model>

saves an archive of the specified model to the specified file. SIMM does not notify you if this action overwrites an existing file.

load <filename>

loads and executes the specified script file. Script files are ASCII text files with lists of commands, with each command on its own line in the file.

pref < name > < string >

sets the preference named *name* to the value *string*.

cd <*folder*>

sets the current folder of the Scripting Tool to the specified folder. This is the folder that is used when opening and saving files. When you open a file by specifying a relative path name, SIMM will look for it in the current folder. When you save a file by specifying a relative path name, SIMM will save it to the current folder. You can use the command "cd .." to move the current folder up one level (e.g., from C:\SIMM\models to C:\SIMM\). When SIMM is first launched, the current folder is set to the value of the preference SCRIPTING_CURRENT_FOLDER.

pwd

prints the name of the current folder in the Messages window.

wait <# *of seconds*>

makes SIMM pause for the specified number of seconds. This can be useful if you are making a script for giving a demo, or for waiting after issuing the *initialize* command in the Dynamics Tool (to give SIMM time to load the simulation DLL in another thread, before issuing the *run* command).

simwait < model>

makes SIMM wait until the dynamic simulation using the specified model has finished. This can be useful if you're making a script that runs a series of simulations, or you want to execute commands only after a simulation has finished (e.g., plotting results).

moviewait < model>

makes SIMM wait until the movie being created is finished writing to the file.

param <model> <param> <value>

sets the specified model parameter (preference) to the specified value. The two parameters that are currently supported are *read script file* and *write script file*.

system <DOS command>

runs the specified DOS command and waits until it is finished. This is useful if, for example, you want to move or make a copy of a data file after outputting it from SIMM.

tool mv

sets the current tool (in the Scripting window) to the Model Viewer. The prompt changes to "[mv]>" to indicate the change.

tool pm

sets the current tool (in the Scripting window) to the Plot Maker. The prompt changes to "[pm]>" to indicate the change.

tool mr

sets the current tool (in the Scripting window) to the Motion Reporter. The prompt changes to "[mr]>" to indicate the change.

tool mt

sets the current tool (in the Scripting window) to the Motion Editor. The prompt changes to "[mt]>" to indicate the change.

tool dt

sets the current tool (in the Scripting window) to the Dynamics Tool. The prompt changes to "[dt]>" to indicate the change.

tool me

sets the current tool (in the Scripting window) to the Muscle Editor. The prompt changes to "[me]>" to indicate the change.

move < window > x y

moves the specified window so that the upper left corner is at screen coordinates (x, y). The origin of this coordinate system is the upper left corner of the SIMM application window, with X going to the right, and Y going down. To specify the window, enter the complete name as shown in the window's title bar, enclosing it in double quotes if it contains spaces.

move <*window*> *x y width height*

moves the specified window so that the upper left corner is at screen coordinates (x, y), and resizes to the specified width and height. The origin of this coordinate system is the upper left corner of the SIMM application window, with X going to the right, and Y going down. To specify the window, enter the complete name as shown in the window's title bar, enclosing it in double quotes if it contains spaces.

refresh <on | off>

turns on or off the refreshing of the SIMM tool windows as script commands are executed. Turning it off will speed up the execution of commands because the tool windows will not be updated and redrawn after each command. Model-based script files generated by SIMM use this feature so that the file can be processed quickly when a model is loaded.

exit

Exits SIMM. SIMM does not prompt you to confirm the action, so make sure you have already saved any desired output files.

2.21.2 Model Viewer Commands

The following commands are available in the Scripting Tool when the Model Viewer is the current tool (enter the command *tool mv* to make the Model Viewer the current tool).

sets the Model Viewer to the specified model. set model <model> restores the camera view of the current model to the one restore camera <name> specified by name. saves the current view of the current model to the camera save camera <name> specified by name. If name is not the name of an existing camera, a new one will be created. restores the pose of the current model to the one specified restore pose < name> by name. save pose <name> saves the current pose of the current model to the pose specified by name. If name is not the name of an existing pose, a new one will be created. turns on or off the display of the specified muscle in the cur*muscles* <*name*> <*on* | *off*> rent model. Name can also be the name of a muscle group. turns on or off the display of the specified muscle menu in *muscle menu* <*name*> <*on* | *off*> the Model Viewer tool *Name* must be the name of a muscle group. animates the current model with the specified motion. The start < motion name > animation begins at the current time value of the motion, and proceeds at the rate specified in the *motion speed* field. start realtime connects the current model to the Cortex or EVaRT program specified running computer the CORTEX MACHINE preference. fixes the origin of the specified body segment in the current look at <segment name> model to the ground, for display purposes only. draw mode <name> <mode> changes the drawing mode of the object specified by *name* in the current model to the mode specified by *mode*. Name can be the name of a body segment, segment group, world object, skin, or muscle surface.

changes the drawing mode of all of the body segments in draw mode all segments < mode> the current model to the mode specified by *mode*. draw mode all world objects < mode> changes the drawing mode of all of the world objects in the current model to the mode specified by *mode*. changes the drawing mode of all of the skins in the current draw mode all skins <mode> model to the mode specified by *mode*. changes the drawing mode of all of the muscle surfaces in draw mode all muscle surfaces <mode> the current model to the mode specified by *mode*. *draw mode all <mode>* changes the drawing mode of all body segments, world objects, skins, and muscle surfaces in the current model to the mode specified by *mode*. turns on or off the slider for the specified gencoord in the gencoords < name > < on | off > Model Viewer tool window. *Name* can also be the name of a gencoord group. turns on or off the sliders for all of the gencoords. gencoords all <on | off> movie file <filename> sets the name of the file to be used when creating a movie of the model window. If "default" is specified as the filename, SIMM will generate an appropriate filename when the movie is created turns on or off auto-movie mode. When auto-movie mode is *auto movie <on | off>* on, each time the model window is refreshed, the image is appended onto the end of the movie. creates a movie of the current model animated by the specimovie <motion name> fied motion. The animation starts at the current time value of the motion, and proceeds at the rate specified in the motion speed field in the Model Viewer. model name < name > renames the current model to the specified name.

motion_speed <percent>

sets the motion speed of the current model to the specified value, which is interpreted as percent of real-time. The value must be between -1000.0 and 1000.0.

trackball<on | off>

turns on or off trackball mode for the current model. See Section 2.4.6, Model Window Viewing Commands, for a description of trackball mode.

muscle_points < *on* | *off*>

turns on or off the highlighted display of muscle points in the current model

shadows <*on* | *off*>

turns on or off the display of shadows in the current model.

crosshairs <on | off>

turns on or off the display of crosshairs on the current model window.

2.21.3 Plot Maker Commands

The following commands are available in the Scripting Tool when the Plot Maker is the current tool (enter the command *tool pm* to make the Plot Maker the current tool).

set model <model>

sets the Plot Maker to the specified model.

set plot <plot>

sets the Plot Maker to the specified plot.

y <*Y*-variable>

sets the Y-variable, which must be one of the following:

musculotendon_length
norm_musculotendon_length
musculotendon_velocity
fiber_length
norm_fiber_length
force
tendon_length
tendon_strain
total_positive_moment
total_negative_moment

muscle orientation

y <*Y*-variable> <*gencoord*>

sets the Y-variable, linked to the specified gencoord. The Y-variable must be one of the following:

moment arm moment_arm_numerical moment max force

x < gencoord >

sets the X-variable to the specified gencoord.

x < motion >

sets the X-variable to the specified motion.

make

executes the make curves command.

delete

deletes the current plot. SIMM does not prompt you to con-

firm the action.

delete <*plot*>

deletes the specified plot. SIMM does not prompt you to

confirm the action.

<muscles> <name> <on | off>

turns on/off the specified muscle for plotting. Name can also

be the name of a muscle group.

<muscle_menu> <group> <on | off>

turns on/off the display of the specified muscle group menu

in the Plot Maker window.

motion_curve <motion> <curve>

plots the specified curve of the specified motion, adding it

to the current plot.

segment <segment name>

sets the segment for the *muscle orientation* Y-variable to the

specified segment.

rename <plot> <new name>

renames the plot specified by plot.

plot_title < title>

sets the name in the plot title field to the specified name.

This title will be used for the next new plot that is created.

sets the name in the *curve name* field to the specified name. curve name < name> This name will be given to the next curve created by the Plot Maker. Enter a name of "null" to make the field empty so that SIMM will generate a default name when a curve is created. sets the step size field to the specified number. step size < number > activation < number> sets the activation field to the specified number, which must be greater than or equal to zero. Setting the activation also turns on the act override checkbox x min < number > sets the *x min* field to the specified number. sets the x max field to the specified number. x max < number > *y min < number>* sets the *y min* field to the specified number. sets the y max field to the specified number. *y max < number>* scale < number > sets the *scale* field to the specified number. offset < number > sets the *offset* field to the specified number.

 $sum < on \mid off > turns on/off the sum checkbox.$

rectify <on | off> turns on/off the rectify checkbox.

active <on | off> turns on/off the active checkbox.

passive <on | off> turns on/off the passive checkbox.

act_override <on | off> turns on/off the act override checkbox.

isometric <on | off> turns on/off the isometric checkbox.

mean calculates the mean and standard deviation of all curves in the current plot. This information is then added to the plot as

a data curve.

2.21.4 Motion Reporter Commands

The following commands are available in the Scripting Tool when the Motion reporter is the current tool (enter the command *tool mr* to make the Motion Reporter the current tool).

set model <model>

sets the Motion Reporter to the specified model.

gencoords < name > < on | off>

turns on/off output to the motion report of the value of the specified gencoord. *Name* can also be the name of a gencoord group, or the word "all".

moments < *name* > < *on* | *off* >

turns on/off output to the motion report of the moment of the specified gencoord. *Name* can also be the name of a gencoord group, or the word "all".

powers < name > < on | off >

turns on/off output to the motion report of the power of the specified gencoord. *Name* can also be the name of a gencoord group, or the word "all".

joint forces < name > < on | off>

turns on/off output to the motion report of the reaction forces of the specified joint. *Name* can also be the word "all".

external forces < name > < on | off>

turns on/off output to the motion report of the external forces acting on the specified body segment. *Name* can also be the word "all".

muscle_activations < *name* > < *on* | *off* >

turns on/off output to the motion report of the activation of the specified muscle. *Name* can also be the name of a muscle group, or the word "all".

muscle_lengths < *name* > < *on* | *off* >

turns on/off output to the motion report of the length of the specified muscle. *Name* can also be the name of a muscle group, or the word "all".

muscle_forces <name> <on | off> turns on/off output to the motion report of the force of the specified muscle. Name can also be the name of a muscle group, or the word "all".
muscle_velocities <name> <on | off> turns on/off output to the motion report of the velocity of the

specified muscle. *Name* can also be the name of a muscle group, or the word "all".

 $motions < name > < on \mid off > turns on/off the specified motion for inclusion in the motion report.$

output_file <filename> sets the output file field to the specified name.

name < name > sets the name field to the specified name.

examiner < name> sets the *examiner* field to the specified name.

study <name> sets the *study* field to the specified name.

height < number> sets the *height* field to the specified number. For most models, the height should be entered in meters.

weight <number> sets the weight field to the specified number. For most models, the weight should be entered in kilograms.

age <age_in_years> sets the age field to the specified number.

id <*value*> sets the *id* field to the specified value, which can be any string of characters.

sets the *test_date* < *day/month/year* > sets the *test_date* field to the specified date. The date should be entered as three numbers with forward slashes between them. Day/month/year is preferred, but month/day/year is acceptable.

gait <on | off> turns on/off the gait checkbox.

standard_deviations < on | off> turns on/off the standard deviations checkbox.

output_strides <on | off> turns on/off the output strides checkbox.

output_norm_strides <on | off> turns on/off the output strides (normalized) checkbox.

generate generates the motion report using all of the current settings.

2.21.5 Dynamics Tool Commands

The following commands are available in the Scripting Tool when the Dynamics Tool is the current tool (enter the command *tool dt* to make the Dynamics Tool the current tool).

set model < model> sets the Dynamics Tool to the specified model.

working_dir < folder> specifies the directory to be used for the dynamic simulation (for input and output files).

dll <filename> specifies the DLL to be used for the simulation.

input_file <filename> specifies the input motion file for the simulation.

output_file <filename> specifies the output motion file created by the simulation.

start time < time > specifies the start time of the simulation.

end_time < *time* > specifies the end time of the simulation.

step size <time> specifies the step size (reporting interval) of the simulation.

motion < motion > specifies which of the motions already linked to the model to use as input to the simulation.

kinematics_cutoff < frequency > specifies the low-pass cutoff frequency to use when smoothing the gencoord data in the input motion. Enter a value of -

1 for no smoothing.

forces_cutoff < frequency > specifies the low-pass cutoff frequency to use when smooth-

ing the external force data in the input motion. Enter a value

of -1 for no smoothing.

specifies the low-pass cutoff frequency to use when smooth-*EMG* cutoff < frequency> ing the EMG data in the input motion. Enter a value of -1 for no smoothing. specifies whether or not to use the gencoord data from the include kinematics < yes | no> input motion in the simulation. specifies whether or not to use the external force data from include forces < yes | no> the input motion in the simulation. specifies whether or not to use the EMG data from the input include EMG < yes | no > motion in the simulation *muscles* <*name*> <*on* | *off*> specifies the muscles to include in the simulation. muscles <group name> <on | off> *muscles all* <*on* | *off*> *normalize* <*none*|1.0|100.0> specifies whether to normalize the output motion to 101 time frames, with the final time being 1.0 or 100.0 seconds. gravity < X|-X|Y|-Y|Z|-Z>specifies the direction of gravity in the simulation. specifies whether or not to use optimization groups when use groups < yes | no> performing static optimization of muscle activations. specifies the value of the Baumgarte constraint stabilization baumgarte < number > parameter, used by the dynamics code to enforce constraints. Number must be greater than or equal to zero. initialize initializes the simulation. runs the simulation, if initialized. run pauses the simulation, if running. pause resumes the simulation, if paused. resume stops and resets the simulation. reset

adds (or removes) the specified gencoord to the list whose gencoord values < name > < on | off > values will be written to the output motion if output of gencoord values is turned on. *Name* can also be the name of a gencoord group, or the word "all". gencoord values <on|off> turns on/off the output of gencoord values. If output is on, only the selected gencoord values will be written to the output motion. gencoord velocities < name > < on | off > adds (or removes) the specified gencoord to the list whose velocities will be written to the output motion if output of gencoord velocities is turned on. Name can also be the name of a gencoord group, or the word "all". turns on/off the output of gencoord velocities. If output is gencoord velocities <on|off> on, only the selected gencoord velocities will be written to the output motion. gencoord accelerations < name > < on | off > adds (or removes) the specified gencoord to the list whose accelerations will be written to the output motion if output of gencoord accelerations is turned on. Name can also be the name of a gencoord group, or the word "all". turns on/off the output of gencoord accelerations. If output gencoord accelerations <on|off> is on, only the selected gencoord accelerations will be written to the output motion. *muscle activations* <*name*> <*on|off*> adds (or removes) the specified muscle to the list whose activations will be written to the output motion if output of muscle activations is turned on. Name can also be the name of a muscle group. *muscle activations* <*on*|*off*> turns on/off the output of muscle activations. If output is on, only the selected muscle activations will be written to the output motion. *muscle lengths* <*name*> <*on|off*> adds (or removes) the specified muscle to the list whose lengths will be written to the output motion if output of

muscle lengths is turned on. *Name* can also be the name of a muscle group. *muscle lengths* < *on* | *off* > turns on/off the output of muscle lengths. If output is on, only the selected muscle lengths will be written to the output motion. adds (or removes) the specified muscle to the list whose *muscle forces* <*name*> <*on|off*> forces will be written to the output motion if output of muscle forces is turned on. *Name* can also be the name of a muscle group. *muscle forces* <*on*|*off*> turns on/off the output of muscle forces. If output is on, only the selected muscle forces will be written to the output motion. *muscle velocities* <*name*> <*on|off*> adds (or removes) the specified muscle to the list whose velocities will be written to the output motion if output of muscle velocities is turned on. *Name* can also be the name of a muscle group. *muscle velocities* <*on|off*> turns on/off the output of muscle forces. If output is on, only the selected muscle forces will be written to the output motion. fiber lengths < name > < on | off > adds (or removes) the specified muscle to the list whose fiber lengths will be written to the output motion if output of muscle fiber lengths is turned on. Name can also be the name of a muscle group. fiber lengths <on|off> turns on/off the output of muscle fiber lengths. If output is on, only the selected muscle fiber lengths will be written to the output motion. adds (or removes) the specified muscle to the list whose *fiber velocities* < name > < on | off > fiber velocities will be written to the output motion if output of muscle fiber velocities is turned on. *Name* can also be the name of a muscle group.

turns on/off the output of muscle fiber velocities. If output is fiber velocities <on|off> on, only the selected muscle fiber velocities will be written to the output motion. contact forces <on|off> turns on/off the writing of contact forces to the output motion. mass center positions <name> <on|off> adds (or removes) the specified body segment to the list whose mass center positions will be written to the output motion if output of mass center positions is turned on. Name can also be the word "all". mass center positions <on|off> turns on/off the writing of mass center positions to the output motion. If output is on, only the mass center positions of the selected segments will be written to the output motion. mass center velocities <name> <on|off> adds (or removes) the specified body segment to the list whose mass center velocities will be written to the output motion if output of mass center velocities is turned on. *Name* can also be the word "all" *mass center velocities* <*on*|*off*> turns on/off the writing of mass center velocities to the output motion. If output is on, only the mass center velocities of the selected segments will be written to the output motion turns on/off the writing of total system energy to the output system energy <on|off> motion. joint reaction forces < name > < on | off > adds (or removes) the specified joint to the list whose reaction forces will be written to the output motion if output of joint reaction forces is turned on. Name can also be the word "all". turns on/off the writing of joint reaction forces to the output joint reaction forces <on|off> motion. If output is on, only the reaction forces of the

selected joints will be written to the output motion.

joint reaction torques <*name*> <*on|off*>

adds (or removes) the specified joint to the list whose reaction torques will be written to the output motion if output of joint reaction torques is turned on. *Name* can also be the word "all".

joint reaction torques < *on* | *off* >

turns on/off the writing of joint reaction torques to the output motion. If output is on, only the reaction torques of the selected joints will be written to the output motion.

gencoord moments <name> <on|off>

adds (or removes) the specified gencoord to the list whose moments will be written to the output motion if output of gencoord moments is turned on. *Name* can also be the name of a gencoord group, or the word "all".

gencoord moments <on|off>

turns on/off the writing of gencoord moments to the output motion. If output is on, only the moments of the selected gencoords will be written to the output motion.

corrected gencoord moments <on|off>

specifies whether or not "corrected" gencoord moments should be written to the output motion. See Section 2.16, Dynamics Tool, for an explanation of the differences between gencoord moments and corrected gencoord moments.

gencoord powers < name > < on | off >

adds (or removes) the specified gencoord to the list whose powers will be written to the output motion if output of gencoord powers is turned on. *Name* can also be the name of a gencoord group, or the word "all".

gencoord powers <on|off>

turns on/off the writing of gencoord powers to the output motion. If output is on, only the powers of the selected gencoords will be written to the output motion.

spring_forces <on|off>

turns on/off the writing of spring forces to the output motion.

num_spring_forces <num>

specifies the number of spring forces to include in the output motion. Space will be made in the motion file to hold

this many spring forces. If more than this number of springs generate force during a time frame of the simulation, only the first *num* of them will be output.

moment arms < name > < on | off >

adds (or removes) the specified muscle to the list whose moment arms for all gencoords will be written to the output motion if output of muscle moment arms is turned on. *Name* can also be the name of a muscle group.

moment arms < *on* | *off* >

turns on/off the writing of muscle moment arms to the output motion. If output is on, only the moment arms of the selected muscles will be written to the output motion.

muscle moments < *name* > < *on* | *off* >

adds (or removes) the specified gencoord to the list whose muscle moments will be written to the output motion if output of muscle moments is turned on. *Name* can also be the name of a gencoord group, or the word "all".

muscle moments < *on* | *off* >

turns on/off the writing of muscle moments to the output motion. If output is on, only the muscle moments of the selected gencoords will be written to the output motion.

total muscle moments < name > < on | off >

adds (or removes) the specified gencoord to the list whose total muscle moments will be written to the output motion if output of total muscle moments is turned on. *Name* can also be the name of a gencoord group, or the word "all".

total_muscle_moments <on|off>

turns on/off the writing of total muscle moments to the output motion. If output is on, only the total muscle moments of the selected gencoords will be written to the output motion.

optimized muscle activations <name> <on|off>

controls which muscles and which gencoords are used in static optimization of muscle activations. *Name* can be the name of a gencoord, gencoord group, or "all_gencoords". It can also be the name of a muscle, muscle group, or "all muscles".

optimized_muscle_activations < on | off>

specifies whether or not the simulation should calculate the muscle activations needed to generate the inverse dynamics moments, and write them to the output motion. If this option is turned on, the activations of every muscle are optimized, but only the moments for the selected gencoords are considered in the optimization.

2.21.6 Motion Editor Commands

The following commands are available in the Scripting Tool when the Motion Editor is the current tool (enter the command *tool mt* to make the Motion Editor the current tool).

set model < model > sets the Motion Editor to the specified model.

set motion < motion > sets the Motion Editor to the specified motion in the current

model.

set event < name > sets the Motion Editor to the event with the specified name

in the current motion.

set event < time> sets the Motion Editor to the event with the specified time in

the current motion.

normalize normalizes the current motion.

normalize < motion > normalizes the specified motion.

smooth smooths the current motion with the default low-pass cut-

off frequency.

smooth < motion > smooths the specified motion with the default low-pass cut-

off frequency.

smooth < motion > < frequency > smooths the specified motion with the specified low-pass

cut-off frequency.

smooth <motion> <tab> <frequency> smooths all of the curves in specified tab of the specified

motion with the specified low-pass cut-off frequency.

smooth <tab> <frequency></frequency></tab>	smooths all of the curves in specified tab of the current motion with the specified low-pass cut-off frequency.
smooth <motion> <curve> <frequency></frequency></curve></motion>	smooths the specified curve in the specified motion with the specified low-pass cut-off frequency.
smooth <curve> <frequency></frequency></curve>	smooths the specified curve in the current motion with the specified low-pass cut-off frequency.
crop	crops the current motion, using the current crop settings.
crop <start_time> <end_time></end_time></start_time>	crops the current motion using the specified start and end times.
crop <motion> <start_time> <end_time></end_time></start_time></motion>	crops the specified motion using the specified start and end times.
crop_start < time>	sets the start time for cropping to the specified time.
crop_end <time></time>	sets the end time for cropping to the specified time.
reset_crop	sets the start and end times for cropping to the start and end times of the current motion.
divide_strides	divides the current motion into strides, creating a new motion with each one.
delete	deletes the current motion. SIMM does not prompt you to confirm the action.
delete <motion_name></motion_name>	deletes the specified motion. SIMM does not prompt you to confirm the action.
delete <event_name></event_name>	deletes the event with the specified name from the current motion. SIMM does not prompt you to confirm the action.
delete <event_time></event_time>	deletes the event with the specified time from the current motion. SIMM does not prompt you to confirm the action.

delete all_motions	deletes all motions in the current model. SIMM does not prompt you to confirm the action.
delete all_events	deletes all events in the current motion. SIMM does not prompt you to confirm the action.
add_event <name> <time></time></name>	adds an event with the specified name and time to the current motion.
show markers <on off="" =""></on>	turns on/off the display of markers in the current motion.
show markers <on off="" =""> <name></name></on>	turns on/off the display of the specified marker in the current motion. <i>Name</i> can also be "all".
show marker_trail_lines <on off="" =""></on>	turns on/off the display of marker trail lines in the current motion.
show marker_trail_lines <on off="" =""> <name></name></on>	turns on/off the display of marker trail lines for the specified marker in the current motion. <i>Name</i> can also be "all".
show marker_trail_points <on off="" =""></on>	turns on/off the display of marker trail lines in the current motion.
show marker_trail_points <on off="" =""> <name></name></on>	turns on/off the display of marker trail lines for the specified marker in the current motion. <i>Name</i> can also be "all".
show force_plates <on off="" =""></on>	turns on/off the display of force plates in the current motion.
show external_forces <on off="" =""></on>	turns on/off the display of all external forces in the current motion.
show body_forces <on off="" =""></on>	turns on/off the display of all body forces in the current motion.
show joint_forces <on off="" =""></on>	turns on/off the display of all joint reaction forces in the current motion.
show joint_torques <on off="" =""></on>	turns on/off the display of all joint reaction torques in the current motion.

turns on/off the display of all contact forces in the current show contact forces <on | off> motion. show spring forces <on | off> turns on/off the display of all spring forces in the current motion *show force trails* <*on* | *off*> turns on/off the display of all force trails (for external forces) in the current motion. turns on/off the display of foot prints in the current motion. show foot prints <on | off> turns on/off the display of gencoord moments in the current *show moments* <*on* | *off*> motion. show other objects <on | off> turns on/off the display of other motion objects in the current motion loop motion <on | off> turns on/off the *loop motion* checkbox for the current motion. turns on/off the enforce constraints checkbox for the current *enforce constraints* < *on* | *off*> motion. *enforce loops* <*on* | *off*> turns on/off the enforce loops checkbox for the current motion

2.21.7 Muscle Editor Commands

The following commands are available in the Scripting Tool when the Muscle Editor is the current tool (enter the command *tool me* to make the Muscle Editor the current tool).

set model <model> sets the Muscle Editor to the specified model.set muscle <muscle> sets the Muscle Editor to the specified muscle.

save all saves all muscles.

save current saves the current muscle.

saves the specified muscle. *save* <*muscle*>

restores all muscles. restore all

restores the current muscle. restore current

restore <muscle> restores the specified muscle.

3 Input Files

3.1 Introduction

A musculoskeletal model is specified with three types of input files. The bone files (Section 3.2) contain lists of polygons that define the bone surfaces. The joint file (Section 3.3) specifies the kinematics of the joints and the characteristics of the body segments. The muscle file (Section 3.4) contains lists of coordinates that describe the muscle lines of action, and the parameters needed to compute isometric muscle forces. SIMM scans these input files and creates a data structure that represents the musculoskeletal model. When SIMM scans the joint and muscle files, it ignores all text between the delimiters /* and */, so you can include comments in the files to help you document your model. Bone files, which cannot contain comments, can be preprocessed with *norm* (see Chapter 4) before being loaded into SIMM.

This chapter describes how to create a musculoskeletal model using input files. Examples from the lower-extremity model are shown to demonstrate the format of each file. The input files that define this model are provided with SIMM. It is often helpful to edit these files and then reload them into SIMM when you are learning to create your own models. The files can also serve as templates when you make a new model.

This chapter also describes motion files (Section 3.5), plot files (Section 3.6), and model archive files (Section 3.7).

Motion files are used to specify the values of the generalized coordinates during a particular motion. They are used to animate the model and to plot data versus a motion parameter. For example, if you define a motion file that specifies the generalized coordinate values during pedaling, you can display the model as it moves through the pedaling cycle and plot independent variables vs. the crank angle. Plot data files are helpful for comparing data from outside sources to data computed with the model. Model archive files are compressed archives that contain all of the files needed for a model.

Note: SIMM does not impose specific units on your musculoskeletal model, nor do you need to define units explicitly. It assumes that the numbers you enter into data files for bone vertices, muscle parameters, and joint kinematics are internally consistent, and it uses them "as is" to calculate dependent variables such as muscle lengths and joint torques. The sample lower-limb model uses units of meters for length and newtons for force.

3.2 Bone Files

In SIMM, a bone file is any file that contains a mesh of planar polygons. Bone files are most frequently used to define anatomical bones that represent a musculoskeletal model's body segments, but they are also used to define world objects, motion objects, force mattes, spring floors, and contact objects. Each bone file can contain descriptions of any number of separate polygonal meshes, or polyhedra, but once loaded into SIMM these polyhedra are treated as a single object. Each polyhedron is usually a closed surface, but it does not need to be closed for SIMM to be able to load and display it (*i.e.*, it can have holes in it). Bone files list the

3D coordinates of the polygon vertices (in a vertex list) and define how the vertices are connected to form polygons (in a polygon list). The reference frame of the body segment to which the bone is attached serves as the local coordinate system for the polygon vertices.

Bone files used as part of a musculoskeletal model can be specified in any of five formats: ASCII, "old" ASCII, binary, Wavefront OBJ, and STL. To include a bone file in a model, put the name of the file in the appropriate object definition in the joint file (see Section 3.3, Joint Files, for details). SIMM will use the file suffix and the data in the first line of the file to automatically determine which format the bone file is in. SIMM can import all of the five formats, but can only output bone files in the ASCII format (see Section 2.15, Bone Editor, for information on writing bone files). SIMM comes with a conversion utility called norm that can also import all five bone file formats, as well as output three of them (ASCII, "old" ASCII, and binary). Norm is located in the directory where the SIMM executable resides. In addition to converting between different file formats, norm processes bone files by calculating vertex normals, checking the vertex and polygon lists inconsistencies, and, optionally, translating and rotating bones and filling holes in their surfaces (see Chapter 4, Norm, for details).

To locate bone files that are referenced in a joint file, SIMM searches the folders in the following list, in the order shown. If a particular folder does not exist or the bone file is not found in it, SIMM proceeds to the next folder in the list.

- (1) the folder specified by the bone_path parameter in the joint file
- (2) the folder *bones* in the folder containing the joint file
- (3) the folder containing the joint file

- (4) the folder specified by the BONES_FOLDER preference (this preference, if not defined by the user, defaults to *Simm\Resources\bones*)
- (5) the folder *Simm**Resources**mocap**bones*

3.2.1 SIMM ASCII

The first bone file format is the recommended format for new musculoskeletal models. It is an ASCII format that was customed developed for SIMM, and an example is shown in Figure 3-1. This format is identified with the *.asc* file suffix.

The first line in the bone file must be the string NORM ASCII. The second line contains the number of vertices (num vertices) followed by the number of polygons (num polygons). The third line contains the bounding box of the bone. The six number are, respectively, X min, X max, Y min, Y max, Z min, and Z max. Then comes the vertex list, whose length must be num vertices. Each line in this list contains the X, Y, and Z coordinates of a vertex and the X, Y, Z coordinates of the vertex normal. The utility program norm can compute vertex normals for you (see Chapter 4, Norm, for more details). Lastly, there is the polygon list, whose length must be num polygons. The first number in each line is the number of vertices (nv) in the polygon, and the other numbers are the indices of the nv vertices. The indices start at 0, and correspond to the order in which the vertices are listed in the vertex list. In the example shown, the first polygon is a triangle, formed by the first, second, and third vertices in the vertex list. The second polygon is four-sided, using the third, fourth, seventh, and sixth vertices. When listing the indices for a polygon, you should order them so that they "walk" around the perimeter of the polygon. You can specify them in either clockwise or counter-clockwise fashion (and can switch

```
NORM ASCII
260
     290
-0.20459 0.45827 -0.35921 0.27346 0.02846 0.68189
-0.062448
           -0.071901
                        0.057002
                                    0.195918
                                               0.980604
                                                           0.005772
                                                          -0.322728
-0.062417
           -0.062953
                        0.055540
                                   -0.010377
                                               0.946434
-0.045919
           -0.070017 0.069408
                                   -0.203832
                                               0.677009
                                                          -0.707184
-0.057682
           -0.053884
                        0.064535
                                    0.232364
                                               0.048436
                                                          -0.971422
           1
           3
                 6
                       5
                 3
                12
                      10
```

Figure 3-1. Example bone file in ASCII format

directions from polygon to polygon), but you cannot list them randomly so that they criss-cross through the polygon. When you process the bone file with *norm*, the polygons will be re-ordered so that all of the vertices are specified in a counterclockwise fashion.

3.2.2 "Old" SIMM ASCII

There is a second version of the SIMM ASCII format that was originally developed for the first version of SIMM. This format is shown in Figure 3-2. Like the first, this format is identified with the .asc file suffix (SIMM examines the first line of text in the file to distinguish between the two formats). It includes less information about the bone, so it is easier to create if you are making a bone file from scratch.

However, if you are converting bone files from another software package, it is recommended that you not convert them to the old ASCII format, but instead convert to OBJ, STL, or the new ASCII format.

The first line in an old SIMM ASCII file must contain the number of vertices (num_vertices) followed by the number of polygons (num_polygons). Then comes the vertex list, whose length must be num_vertices. Each line in this list contains the X, Y, and Z coordinates of a vertex. Lastly, there is the polygon list, whose length must be num_polygons. The first number in each line is the number of vertices (nv) in the polygon, and the other numbers are the indices of the nv vertices. The indices start at 1, and correspond to the order in which the vertices are listed in the vertex list. In the example shown, the first polygon is a tri-

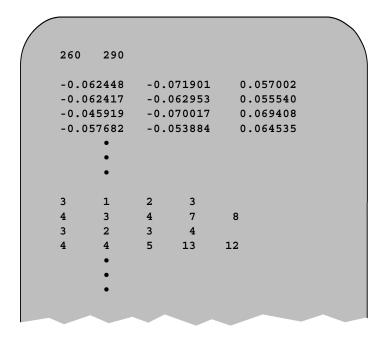


Figure 3-2. Example bone file in "old" ASCII format

angle, formed by the first, second, and third vertices in the vertex list. The second polygon is four-sided, using the third, fourth, seventh, and eighth vertices.

Note: when a musculoskeletal model is loaded into SIMM, all of the model's bones that are specified in the old ASCII format are automatically processed with *norm* so that the vertex normal information is available. For bones with many polygons, this can add significantly to the time it takes to load a model. Thus for large bone files it is recommended that you convert them to the [new] SIMM ASCII format before adding them to your model.

3.2.3 SIMM Binary

The third format supported by SIMM and *norm* is a binary format. The information in the file is identical to that in the old SIMM ASCII format, but the file is binary in order to reduce its size. This format is identified with the *.bin* file suffix. Previous versions of SIMM required bones to be in this binary format, so corresponding versions of *norm* would output binary bone files by default.

Note: when a musculoskeletal model is loaded into SIMM, all of the model's bones that are specified in the SIMM binary format are automatically processed with *norm* so that the vertex normal information is available. For bones with many polygons, this can add significantly to the time it takes to load a model. Thus for large bone files it is recommended that you convert them to the SIMM ASCII format before adding them to your model.

3.2.4 Wavefront OBJ

SIMM also supports bone files in the Wavefront OBJ format, identified with the .obj file suffix. This is an ASCII format developed by Wavefront Technologies that contains positions and normals of the vertices, plus how they are connected into polygons. Comments can be included in the file by beginning the line with a #. An example is shown in Figure3-3. OBJ files can also contain UV texture coordinates for each vertex, as well references to files with material and texture information. However, these material and texture data are currently ignored by SIMM. A complete description of the OBJ format can be found on various web sites, including: http://en.wikipedia.org/wiki/Obj.

Note: when a musculoskeletal model is loaded into SIMM, all of the model's bones that are specified in the Wavefront OBJ format are automatically processed with *norm* to

Figure 3-3. Example bone file in Wavefront OBJ format

ensure that the vertex normal information is complete. For bones with many polygons, this can add significantly to the time it takes to load a model. Thus for large bone files it is recommended that you convert them to the SIMM ASCII format before adding them to your model.

3.2.5 STL

SIMM also supports bone files in both the ASCII and binary STL formats (both are identified with the .stl file suffix). This is a format originally developed for stereolithography CAD systems, but is also commonly used when reconstructing 3D objects from 2D medical image data. An example of the ASCII STL format is shown in Figure 3-4. A polyhedron is specified as a series of unconnected facets (usually

```
solid
facet normal -0.432299 0.719353 -0.543736
  outer loop
    vertex 123.269600 79.752400 1.403651
    vertex 123.702500 79.963800 1.339151
    vertex 123.299000 79.632500 1.221651
  endloop
endfacet
facet normal 0.016157 0.316683 -0.948394
  outer loop
    vertex 123.702500 79.963800 1.339151
    vertex 124.409800 79.707000 1.265451
    vertex 123.299000 79.632500 1.221651
  endloop
endfacet
endsolid
```

Figure 3-4. Example bone file in ASCII STL format

triangles), each with a normal vector. A complete description of the STL format can be found on various web sites, including: http://en.wikipedia.org/wiki/STL_(file_format).

Note: when a musculoskeletal model is loaded into SIMM, all of the model's bones that are specified in the STL format are automatically processed with *norm* so that the facets (polygons) can be connected together and vertex normal information is available. For bones with many polygons, this can add significantly to the time it takes to load a model. Thus for large bone files it is recommended that you convert them to the SIMM ASCII format before adding them to your model.

3.2.6 Collada

SIMM also supports bone files in the Collada format (.dae). Collada is an open-standard XML format that is supported by many 3D graphics and animation software packages. SIMM will read polygonal mesh, material, and texture map definitions from each Collada file and use them to create one bone for each file. If vertex normals are not specified in the file for the polygonal meshes, then SIMM will post-process the Collada bone with *norm* in order to calculate them. Texture maps within Collada files must be specified with JPG or TIF files.

3.2.7 IGES

SIMM also supports bone files in the IGES format. IGES (Initial Graphics Exchange Specification) is an open-standard text format that is commonly used to transfer designs between CAD software packages. SIMM will read NURBS surfaces from IGES files by importing NURBS patch definitions as well as trimming curves for the patches. NURBS

surfaces are required in SIMM to create NURBS contact and NURBS/spring contact objects; these cannot be created from polyhedra. To display NURBS surfaces in the model window, SIMM tesselates them into a set of triangles. In some cases the tesselation may look rough or contain small gaps between the patches, but the triangles are used for display only. The original NURBS patch definitions are used when creating contact objects to be used in dynamic simulations. There is currently no way to output IGES files from SIMM, but bones in SIMM that came from IGES files can be exported as polyhedra (SIMM can output the tesselation triangles to ASC, BIN, OBJ, or STL files).

3.3 Joint Files

Joint files define the body segments, joints, generalized coordinates, and kinematic functions used in musculoskeletal models. They can also include optional components such as world objects, muscle wrap objects, deform objects, constraint objects, and materials. Each model you load into SIMM must have one joint file defining all of the segments, joints, generalized coordinates, and kinematic functions that comprise the model. You may list these components in the file in any order, but the order in which you enter segment names can affect the display of the model by influencing the choice of the *fixed* segment (see Section 3.3.2, Fixed Segment, for more information).

The joint file also contains a number of parameters that specify various properties of the model, such as its name, some of the colors used in the display, and the units of length and force in which the model is specified. All of these parameters are optional, and have reasonable default values if they are not specified in the joint file. These

parameters generally go at the top of the file, although they can be placed anywhere. The rest of this section describes these optional parameters.

Note: Case is significant when entering text into joint files. All SIMM keywords (e.g., name, beginsegment) must be in lower-case letters. You are free, however, to use upperas well as lower-case letters for naming your model components (e.g., names of joints and segments), and SIMM will display these names exactly as you type them into the joint file.

name

To give your model a name, use the keyword name followed by the desired name. SIMM assumes that all text following the keyword on the same line comprises the name, so you can put spaces in your model name. If you do not name your model in this way, SIMM will assign a default name to it, consisting of the word "model" followed by a number. Here is an example:

name Lower Extremity

muscle file

This parameter specifies the name of the muscle file that will be loaded with the joint file. This is a convenient feature if you often load the same muscle file with a certain joint file. To specify the muscle file, use the keyword muscle_file followed by the name of the file. Then when you load a model by selecting the joint file with the file browser, SIMM will also open the specified muscle file. This parameter can be changed within SIMM by right-clicking on the background in the model window and choosing edit model preferences from the pop-up menu.

muscle file leg model.msl

motion file

This parameter specifies the name of the motion file[s] that will be loaded with the joint file. To specify a motion file, use the keyword motion_file followed by the name of the

file. Then when you load a model by selecting the joint file with the file browser, SIMM will also open the specified motion file[s] and link them to the model. You can specify up to 100 motion files in each joint file.

```
motion_file walk.mot
motion file run.mot
```

bone_path

This parameter specifies the folder containing the bone files used in the model. SIMM will look in this folder first to find each bone file, but if a particular file is not found SIMM will search in other folders as well. Thus SIMM might open some bone files from one folder, and other ones from another folder. The complete list of folders SIMM will search for bone files is given in Section 3.2, Bone Files. This parameter can be changed within SIMM by right-clicking on the background in the model window and choosing *edit model preferences* from the pop-up menu.

```
bone path ..\bones
```

read_script_file

If this parameter is on, SIMM will read a script file each time it loads the model. This file must be in the same folder as the joint file, and must have the same base name with the suffix ".ssc". This file can contain any SIMM script commands, but it is designed to contain commands to specify settings in the tools. It is particularly useful for specifying the output variables for the Motion Reporter and Dynamics Tool so that you do not need to reselect them each time you load the model. This parameter can be changed within SIMM by right-clicking on the background in the model window and choosing *edit model preferences* from the popup menu.

```
read script file on
```

Note: model archives contain their corresponding script files within the archive. If there is a separate script file in the same folder as the archive, it will not be read when the model is loaded.

write script file

If this parameter is on, SIMM will write a script file each time the model is saved or closed. The file is created in the same folder as the joint file, and has the same base name with the suffix ".ssc". The file contains commands to restore the settings in the Model Viewer, Plot Maker, Motion Reporter, Motion Editor, and Dynamics Tool to the same values as when the file was created. This parameter can be changed within SIMM by right-clicking on the background in the model window and choosing *edit model preferences* from the pop-up menu.

read_script_file on

Note: model archives contain their corresponding script files within the archive. When you save an archive, its script file will be updated and stored within it. When you close a model that came from an archive, the script file inside the archive is not updated with the latest tool settings.

inverse kinematics solver

If there are loops (closed kinematic chains) in your model, you can use this parameter to specify whether the inverse kinematics solver is on or off (it is on by default). Specify the desired state after the keyword: on or off. If the solver is on, and there are loops in the model, SIMM will try to assemble the model so that all the loops are closed, using the default configuration as a starting point. See Section 2.8.5, Inverse Kinematics, for more information.

inverse_kinematics_solver on

background color

This parameter specifies the background color of the model window. The RGB components are in the range 0.0 to 1.0. The default color is 0.2 0.2 0.2.

background_color 0.2 0.2 0.2

rotation axes color

This parameter specifies the color of the vectors used to represent the joint rotation axes in the model window. The RGB components are in the range 0.0 to 1.0. The default color is 1.0 1.0 0.0.

background color 1.0 1.0 0.0

segment axes color

This parameter specifies the color of the reference frames of the body segments in the model window. The RGB components are in the range 0.0 to 1.0. The default color is 1.0 1.0 1.0.

background_color 1.0 1.0 1.0

vertex label color

This parameter specifies the color of the polygon outline and the vertex numbers (used by the Muscle Editor) in the model window. The RGB components are in the range 0.0 to 1.0. The default color is 1.0 0.0 0.0.

background_color 1.0 0.0 0.0

crosshairs color

This parameter specifies the color of the crosshairs shown in the model window (when the crosshairs checkbox is selected in the Model Viewer). The RGB components are in the range 0.0 to 1.0. The default color is 1.0 1.0 1.0.

background_color 1.0 1.0 1.0

length_units
force units

These parameters specify the length and force units of the model, which are used for two purposes. First, the Motion Module uses them to convert 3D marker coordinates and forceplate data in TRB/C, ANB/C, and C3D files into SIMM model coordinates. Second, the units are used for display purposes when creating plots. In this case, SIMM does not use the units to convert any data values, it merely adds the unit labels to the plot labels. Length_units specifies the units of all lengths and translations in your model. Force_units specifies the units of muscle force. These parameters can be changed within SIMM by right-clicking

on the background in the model window and choosing *edit model preferences* from the pop-up menu.

length_units m
force units N

window position

This parameter specifies the position of the model window when the model is first loaded into SIMM. The X and Y coordinates are relative to the top left corner of the main SIMM window, with X going to the right, and Y going down.

window_position 800 20

window size

This parameter specifies the size of the model window when the model is first loaded into SIMM. The numbers following the keyword are the width and height, respectively.

window size 400 600

gravity

This parameter specifies the direction of the gravity vector, which is used only during dynamic simmulations. The default gravity vector is -Y. Available options are: -X, +X, -Y, +Y, -Z, +Z.

gravity -Z

solver_accuracy
solver_max_iterations
solver_method

There are three parameters you can specify to control the accuracy of the least-squares solver module within SIMM. This module is used by the Motion Module to solve frames of marker data, and by SIMM to implement closed loops and constraint objects. For both uses, the module adjusts the model's gencoords iteratively until the markers, loops, and/ or constraints (collectively called "solver ordinates") are satisfied to within the specified accuracy, or until the specified maximum number of iterations has been reached. The accuracy value corresponds to the sum of the squares of the errors between the current solver ordinates and the "ideal"

ordinates. Decreasing this value will cause the solver module to iterate longer and return a more accurate solution. The solver_method allows you to choose between the Levenberg-Marquardt method (specified by LM) and the Gauss-Newton method (specified by GN) of least-squares solving. The latter method is usually faster, but can be less robust when some marker data is missing. Shown below are the parameters with their default values.

solver_method LM
solver_accuracy 0.0001
solver max iterations 100

loop tolerance

This parameter specifies the "acceptance level" for the solutions that the solver module reaches when implementing the loop constraints. It does not affect the speed or accuracy of the solver module. Rather, it is a cutoff value specifying whether or not the solution is acceptable. If it is not acceptable, a dialog box may be presented indicating that the loop constraint is not satisfied (this notice appears only when loading or editing a model, not when changing gencoord values). This tolerance value corresponds to the distance (in model units) between the two segments of the loop joint (see Section 3.3.3, Joints, for an explanation of the loop joint). When the loop is properly enforced, the points on the two segments are coincident, and the distance is zero. When the loop is not properly enforced, the loop joint disarticulates and the two points are separated by a certain distance. The loop tolerance parameter specifies how large a distance is acceptable. Shown below is the parameter with its default value. Once the model has been loaded into SIMM. this parameter can be changed via the Gencoord Editor.

loop tolerance 0.0001

marker_visibility
 marker_radius

These parameters specify properties of all of the markers in the model. Marker_radius is the radius of the spheres used to represent the markers in the model window. It affects only their display; it does not affect any marker offset calculations that the Motion Module performs when loading the model. Marker_visibility controls the visibility of all of the markers in the model window. It is a convenience feature that allows you to turn all of the markers on or off without selecting them individually and changing their visibility. Even if the markers are not visible, they are still used by the Motion Module when importing tracked marker files. Both of these marker parameters can also be changed using the Marker Editor. Shown below are the parameters with their default values:

marker_visibility no
marker radius 0.005

3.3.1 Body Segments

A body segment consists of one or more bones fixed in a reference frame. The only required component of a segment definition is a name. You will usually want to specify one or more bone files as well, however, so that you can see the body segment in the model window. Other optional components include the segment's material properties, drawmode, shadow, shadow color, associated deform objects, and display of the segment's reference frame. (Figure 3-3).

Bones are specified using the keyword bone, followed by the name of the bone file, and, optionally, the material and drawmode for displaying the bone. The material is used to set the color and lighting of the bone. The drawmode gives you control over the shading of the bone's surface. These optional drawmode and material identifiers can be placed in any order after the file name. If you do not specify the material or drawmode for a bone, it will inherit the material or drawmode of the segment (as specified with the drawmode and material keywords). If no material is assigned to the

body segment, the default material, def bone, is used. See Section 3.3.7, Materials, for information on defining a If no drawmode is defined, the default is gouraud shading. There are six drawmodes available: wireframe, solid fill, flat shading, gouraud shading, outlined, and none. Wireframe means draw just the outlines of the polygons in the bone. Solid fill means draw the bone as a collection of filled polygons. For both of these drawmodes, the color used to draw the bone is the ambient color specified in the bone's material (see Section 3.3.7, Materials). Flat shading means use the light sources in the model window to shade the bone polygons, but fill the polygons using only one color each. This drawmode will accentuate the individual faces on the bone surface. Gouraud shading means to smooth-shade each polygon. This is often the slowest drawmode because it uses the light sources to calculate the color of each pixel on the bone surface so that it is rendered smoothly, hiding the polygon edges. Outlined means to draw the polygons in the bone's normal material, and highlight the edges in a darker color. None means to not display the body segment at all. You can also interactively change the drawmode of any bone or world object using the drawmode command in the Model Viewer, or by right-clicking on the bone or world object in the model window. If you interactively change the material or drawmode, the new values will be written to the joint file when you save the model.

The begingroups and endgroups keywords can be used to enclose the name(s) of one or more body segment groups to which the segment belongs. Organizing your body segments into groups can make it easier to navigate and display your model with the SIMM Model Viewer.

The shadow parameter is used to specify the direction and location of the body segment's shadow. In the example

```
/* BODY SEGMENT: right lower leg */
beginsegment right shank
                               /* name of this body segment */
bone r tibia.asc wireframe
                               /* this bone uses the default material */
bone r fibula.asc flat shading my mat /* this bone overrides the default */
                                      /* material and drawmode */
begingroups
right leg lower body
                                 /* this segment belongs to two groups */
endgroups
drawmode flat shading
                                /* override default and use flat shading */
                                /* override default, which is def bone */
material my bone mat
shadow Y -0.9
                                /* draw the shadow directly below */
shadowcolor 0.1 0.1 0.1
                                /* use a dark gray for the shadows */
                                /* show reference frame, w/axes 0.2 long */
axes 0.2
                                /* mass (in kg) */
mass 3.25
masscenter 0.0 -0.18 0.0
                                 /* location of mass center */
                                /* inertia matrix relative to mass center */
inertia 0.04 0.0
                   0.0
        0.0 0.006 0.0
                                       and axes of segment reference frame */
        0.0 0.0 0.03
marker R.Ankle.Lat -0.005 -0.410 0.053 2.0 /* weight
marker R.Ankle.Med -0.005 -0.410 -0.053 2.0 /* weight
marker R.Shank 0.02 -0.25 0.0 1.0 fixed /* weight = 1, fixed in place */
gait scale R SHANK 0.39 0.39 0.39
                                            /* used for scaling generic model */
endsegment
```

Figure 3-3. Example body segment

shown in Figure 3-3, the shadow of the femur would be displayed at a Y-value of -0.9 meters from the origin of the world reference frame, and would look as if the light casting the shadow were positioned on the Y-axis. The shadow-color parameter can be used to specify the red, green, and blue components, respectively, of the shadow's color. Each of the components can range in value from 0 to 1.0, and the default color is 0.08, 0.08, 0.08. To display a body segment's shadow in the model window, use the **shadows** toggle button in the Model Viewer (see Section 2.4.5, Toggle Buttons). When you turn shadows on, only those body segments for which you have specified shadow properties will cast a shadow.

The axes parameter is used to display the coordinate system of the body segment in the model window. The number after the keyword axes is the length of each vector forming the reference frame.

To use the model with the Motion Module or the C3D Module for importing tracked marker data, you must also specify markers on some of the body segments. A marker is specified with the keyword marker followed by its name. Marker names can contain spaces, so the name must be followed by a tab, and then the XYZ location of the marker in the body segment's reference frame. The fourth number is marker's weight in the marker tracking algorithm. After the weight you can optionally specify whether the marker is fixed to the body segment (fixed) or whether its location can be adjusted by the Motion Module, and whether it is visible or not in the model window (visible, invisible). To use the model with the Motion Module's subject-specific scaling features, you must also specify scaling parameters for the segments. These are specified with the gait scale keyword followed by the name of the reference segment and the segment's XYZ scale factors relative to the reference segment. Read Chapter 5, Motion Module, for details on how to specify these parameters and how they are used.

To use the model in dynamic simulations with the Dynamics Pipeline or FIT Module, you must also specify the inertial parameters of each body segment. Use mass to specify the mass, and masscenter to specify the XYZ location of the mass center in the body segment's reference frame. The inertia matrix is specified with inertia, and the values should be with respect to the mass center of the segment and relative to the segment's reference frame.

3.3.2 Fixed Segment

The fixed segment is the body segment that does not move in the SIMM display when you move the joints in your musculoskeletal model. That is, the origin of this segment is fixed, but the segment is allowed to rotate about its origin (if the joints in your model allow this). If you define a segment named "ground," then this segment is automatically marked as the fixed segment. Otherwise, the segment whose name appears first in the joint file is marked as the fixed segment. Note that segment names are used in joint definitions as well as segment definitions, so the order in which you enter joints may also matter. For example, in the sample lowerextremity file supplied with SIMM, pelvis is the first body segment referenced in the joint file, so that segment is the fixed segment. When you flex, adduct, or rotate the hip, the femur rotates with respect to the pelvis, which remains fixed in the world reference frame. If the joint file were altered so that the femur were referenced first, then the pelvis would rotate with respect to the femur.

If you explicitly define a ground segment, you can define its characteristics just like you can with any other body segment. You can specify one or more bone files so that you can see the ground segment in the model window.

You can interactively change the fixed segment using the **look at** command in the Model Viewer (see Section 2.4.2, Command Menu, for details). When you select a body segment with this command, that segment becomes the new fixed segment. If you write the model to a file, however, this new fixed segment is not marked as such in the output file.

3.3.3 Joints

The body segments can be connected in any arrangement by defining joints. A joint specifies the transformations that relate the position and orientation of one body segment to another. The body-fixed transformations are performed in a user-defined order, and consist of three orthogonal translations and three rotations around user-defined axes. "Body-fixed" means that the axes are fixed in the child body (the body that the joint is "to"). When each joint rotation is performed, the rotation and translation axes are rotated with it. Thus the next rotation is performed about the newly rotated axis. This is in contrast to "space-fixed," in which all of the rotation and translation axes are fixed in the parent body.

Each translation and rotation in a joint is either constant (*i.e.*, independent of body position) or a function of a generalized coordinate. A generalized coordinate defines a degree of freedom in the musculoskeletal model (*e.g.*, a joint angle). The functions that relate the translations and rotations to the generalized coordinates are called kinematic functions. A sample joint definition is shown in Figure 3-4.

The translations and rotations that comprise a joint are called *dofs*. The rotation dofs (r1, r2, r3) are rotations around axis1, axis2, and axis3, respectively. The translation dofs (tx, ty, tz) are translations along the X, Y, and Z axes (respectively) of the first body segment listed after the segments keyword. The second body segment listed is the one on the other side of the joint (the reference frame you end up after doing the ordered transformations). To make any of the dofs a constant, put the keyword constant after the name of the dof, then enter a value. This value should be in degrees for the rotations. For translations you can use any units you like, but they must match the units used in the bone and muscle files (meters are used in the sample leg

model). To make a dof a function of some generalized coordinate, put the keyword function after the name of the dof, then enter a function name, followed by the name of a generalized coordinate, in parentheses. The function name must start with the letter f, followed by a number.

In the example in Figure 3-4, the femoral-tibial joint definition (part of the knee joint) specifies the transformations from the femur segment to the tibia segment. The order of the transformations indicates that to move from the femur reference frame to the tibia reference frame, a translation is performed first, followed by rotations around axis1, axis2, and axis3, respectively. The translation is divided into three orthogonal components (tx, ty, tz), where the X, Y, and Z directions are defined by the axes of the femur reference frame. The rotation axes are also defined relative to the femur reference frame, and their orientations can be

```
/* FEMORAL-TIBIAL JOINT */
beginjoint femoral-tibial
                              /* name of this joint */
                              /* joint from FEMUR frame to TIBIA frame */
segments femur tibia
                              /* translate, then rotate about axis1, axis2,
order t r1 r2 r3
                                  and axis3 */
axis1 0.0 0.0 1.0
axis2 1.0 0.0 0.0
                              /* rotation axis definitions */
axis3 0.0 1.0 0.0
loopjoint
show axis1 0.2
                              /* display rotation axis1, length = 0.2 */
tx function f1 (knee angle)
                              /* The kinematic functions f1 and f2 */
ty function f2 (knee angle)
                              /* specify the relations between the */
tz constant 0.0
                              /* generalized coordinate (knee angle) */
r1 function f3 (knee angle)
                              /* and tx and ty, the translations in */
r2 constant 0.0
                              /* the x and y directions, respectively. */
r3 constant 0.0
                              /* f3 specifies the relation between */
endioint
                              /* knee angle and the rotation around axis1. */
```

Figure 3-4. Example joint definition

changed by adjusting the axis definitions. Since all of the translations and rotations in this joint are either constants or functions of a single generalized coordinate (knee_angle), the joint has only one degree of freedom. It is not a simple revolute, however, because two of the translations (tx, ty) are functions of knee_angle.

If you want to display any of the axes of rotation in the model window, put the keyword show in the joint definition, followed by the name of the axis (i.e. axis1, axis2, or axis3) and the desired length of the axis. This can be helpful when verifying a new joint definition or when visualizing moment arm calculations.

The keyword loopjoint should no longer be needed in a joint definition, now that SIMM includes an inverse kinematics (IK) solver. If you have loops of body segments in your model, the IK solver will find gencoord values that satisfy the loop constraints and keep the loops closed. If you turn the IK solver off, or if it cannot find a configuration which keeps the loops closed, SIMM will display your model with one of the joints in the loop broken. This socalled "loop joint" will be ignored for display and plotting purposes, so you won't be able to animate the joint or plot muscle properties as a function of the joint's gencoords. To specify which joint should act as the loop joint, put the keyword loopjoint in the joint definition. In most cases, however, it is recommended that you do not specify loop joints, and instead allow the IK solver to control the loops in your model.

3.3.4 Generalized Coordinates

Generalized coordinates are degrees of freedom in the musculoskeletal model. The definition of the generalized coordinate knee angle, from the femoral-tibial joint definition, is shown in Figure 3-5. A range of motion is defined that limits the values of the generalized coordinate and serves as the independent variable when making plots. If the generalized coordinate represents a rotational degree of freedom, the range should be specified in degrees. If the gencoord is translational, the units should match the units of the other lengths and translations that you specify in your model. Also specified is the keyboard key (k key) that controls the value of the generalized coordinate when animating the model in the model window. You can specify either one or two keys to control the generalized coordinate, and they must be on the same line as the word keys in the joint file. The keyword wrap is an optional parameter that affects the control of the generalized coordinate in the Model Viewer. When you include wrap in the definition, you can repeatedly sweep through the range of motion for the generalized coordinate by resetting to the beginning of the range when you reach one end. This is described in more detail in Sec-

```
/* Generalized Coordinate: knee angle */
                             /* name of gencoord */
begingencoord knee angle
begingroups
right leg lower body
                             /* this gencoord is a member of two groups */
endgroups
range
       0.0 -120.0
                              /* range of motion for this gencoord (degrees) */
                             /* press this key to change gencoord value */
keys k key
default value -30.0
                             /* default (and initial) value of gencoord */
                             /* lets you loop through range of motion */
restraint f19
                             /* function used to keep gencoord in range */
                             /* is restraint function currently active? */
active yes
                             /* tolerance (in degrees) */
tolerance 0.25
                             /* is gencoord clamped to range of motion? */
clamped yes
                             /* is gencoord locked at current value? */
locked no
                             /* is gencoord slider visible in Model Viewer? */
visible no
pd stiffness 0.5
                             /* stiffness for PD controller during dynamics */
endgencoord
```

Figure 3-5. Example generalized coordinate

tion 2.4, Model Viewer. To define a default value for a generalized coordinate, use the keyword default_value followed by some number in the range of motion. This will be the starting value of the generalized coordinate, and the value it will be reset to when you select **reset gc values** in the Model Viewer.

The keyword clamped is an optional parameter that lets you specify whether or not the gencoord is constrained to remain within its specified range. By default, gencoords are clamped and cannot assume values outside their range. The keyword locked is an optional parameter that lets you specify whether or not the gencoord is locked. By default, gencoords are unlocked. If a gencoord is locked, its value cannot change.

Restraint functions can be specified for each gencoord. These are used by Dynamics Pipeline simulations to apply restraint torques to prevent the gencoords from moving beyond their ranges of motion. They are also used by the inverse kinematics solver to assign weights to the gencoords when finding model configurations to satisfy the loop constraints. A restraint function is specified using the keyword restraint and a function name in the same format as that used for kinematic functions.

The keyword active is an optional parameter that can be used when a restraint function is associated with the gencoord. If the function is active, it will be used by the inverse kinematics solver when trying to close loops. If the function is inactive, it has no effect. This is useful if you want to use the function in the Dynamics Pipeline, but don't want to use it to solve loops. If you define a restraint function for a gencoord, it is active by default.

The keyword pd_stiffness can be used to specify a stiffness value that is used by the Dynamics Pipeline when cal-

culating *joint torques w/corrections* while performing a dynamic simulation. This calculation uses a PD (proportional-derivative) controller to determine the torques necessary to drive a model to follow the input motion. Consult the Dynamics Pipeline manual for more information about this feaure.

The keyword tolerance can be used to specify a tolerance value used when the gencoord is set to a new value. If the difference between the new value and the current value is less than the tolerance, the new value is ignored, and the gencoord remains unchanged. For most applications you do not need to specify a tolerance (the default is 0.0). However, it can be useful when you are importing a motion which contains identical or similar gencoord values for some periods of time. By setting the tolerance of each gencoord to, say, 0.5 degrees, small changes in gencoord values will not result in the joint transformation matrices being recalculated, thus saving some computation time. This feature is especially useful when you are importing a motion in real-time (with the Motion Module), and want the SIMM animation to be as fast as possible.

By default, the slider for each gencoord will be visible in the Model Viewer window. If you want to hide the slider for a particular gencoord, use the visible keyword with the value no. Gencoord sliders that are initially hidden when the model is loaded can be made visible using the **gencoords** command in the Model Viewer.

The begingroups and endgroups keywords can be used to enclose the name(s) of one or more gencoord groups to which the gencoord belongs. Organizing your generalized coordinates into groups can make it easier to navigate and display your model with the Model Viewer.

3.3.5 Functions

Functions are used in SIMM for a variety of purposes, such as constraining the motion within a joint (kinematic functions), limiting the range of motion of the gencoords (restraint functions), specifying excitation patterns for muscles (excitation functions), and specifying the movement of a muscle point relative to its body segment (moving muscle point functions). Each function is defined by a series of XY control points in parentheses (e.g., (5.7, -12.6)). A comma separates the two components, and you can put spaces between the values, commas, and parentheses. The control points are interpolated using one of four methods: step function, [piecewise] linear function, natural cubic spline, and GCV spline.

A step function, also called piecewise constant or zero-order hold, has a slope of zero at all points, and is represented by a series of intervals at the specified Y values. For X values less than the first XY control point (X_0, Y_0) , the function evaluates to Y_0 . The definition of a step function must have at least two control points, and looks like this:

```
beginstepfunction (-20.0, 0.5) (0.4, 4.3) (10.8, 9.7) endstepfunction
```

A piecewise linear function is a first-order interpolation of the control points (*i.e.*, they are connected by straight lines). For X values beyond the range of the XY control points, the slope of the function at the nearest endpoint is used for extrapolation. The definition of a linear function must have at least two control points, and looks like this:

```
beginlinearfunction
(-22.0, 0.5)
```

```
(0.7, 4.5)
(10.0, 9.7)
(35.4, 5.8)
endlinearfunction
```

A natural cubic spline is a third-order piecewise polynomial interpolation of the control points. It has a continuous second-order derivative and thus is a good candidate for kinematic and restraint functions. For X values beyond the range of the XY control points, linear extrapolation is used, using the slope of the function at the nearest endpoint. The definition of a natural cubic spline must have at least two control points, and looks like this:

```
beginnaturalcubicspline
(-20.0, 0.5)
(0.0, 4.2)
(12.0, 9.7)
(35.5, 5.8)
endnaturalcubicspline
```

Note: In earlier versions of SIMM, natural cubic was the only type of function supported. Thus, if an old-style function definition is used (beginfunction, endfunction), a natural cubic spline is assumed. Also, although a natural cubic spline is a third-order function and thus requires at least four control points, earlier versions of SIMM allowed definitions with as few as two points. If three points were specified then quadratic interpolation was used, and if two points were specified then linear interpolation was used. The current version of SIMM maintains backwards compability by allowing the same function specifications.

A GCV spline is a third-order function that uses the generalized cross-validation and mean-squared prediction error criteria developed by Craven and Wahba, and implemented by Woltring. It has a continuous second-order derivative and thus is a good candidate for kinematic and restraint func-

tions. It also has support for smoothing, a feature used by the Dynamics Pipeline to smooth data input to inverse dynamics simulations. The definition of a GCV spline must have at least six control points, and looks like this:

```
begingcvspline
(-20.0, 0.5)
(-10.0, 2.5)
(0.0, 4.3)
(10.5, 9.7)
(23.7, 8.2)
(35.0, 5.8)
endgcvspline
```

Kinematic Functions

Kinematic functions define translations and rotations between body segments as functions of the generalized coordinates. For example, the anterior-posterior translation (tx) between the femur and the tibia is often modeled as a function of knee angle, as shown in Figure 3-6. The (knee angle, tx) pairs are interpolated by a natural cubic spline. When defining a kinematic function, you should make sure that you define it over a large enough X range so that it is defined over the complete range of motion of every generalized coordinate that uses it. If you do not, SIMM will extrapolate the function, but this might not give the desired results (depending on the type of function). The X coordinates, like the range of motion of gencoords, should be specified in degrees for rotational gencoords. For translational functions, you should use units that are consistent with the other lengths and translations that you specify in your model (meters, for most models).

Restraint Functions

Restraint functions are used primarily for controlling gencoords during forward dynamic simulations performed with the Dynamics Pipeline. The functions define forces or torques that are applied to the gencoords in order to restrain them from going outside their ranges of motion. However, restraint functions are also used by SIMM's inverse kinematics (IK) solver when you have one or more closed loops of body segments. In this case, the restraint functions define error values that restrain the gencoords from going outside their ranges of motion while the IK solver is satisfying loop constraints (see Section 2.8.5, Inverse Kinematics, for more details on the IK solver).

If there are no loops of body segments in your model, the IK solver will be inactive, and the gencoords' restraint functions will not be used in SIMM. If your model does have loops, you may want to define restraint functions to better control the IK solver. If you do define restraint functions for use with IK, you should define them so that they also have appropriate dynamic behavior, and thus can be used with the Dynamics Pipeline as well.

```
/* KINEMATIC FUNCTION: tx of the femoral-tibial joint */
beginnaturalcubicspline f1 /* defines x translation from femur to tibia */
/* knee_angle (degrees), tx (meters) */
(-120.0,-0.003)
(-100.0, 0.001)
( -80.0, 0.004)
( -60.0, 0.004)
( -40.0, 0.002)
( -20.0,-0.001)
( -10.0,-0.003)
(  0.0,-0.005)
endnaturalcubicspline
```

Figure 3-6. Example kinematic function

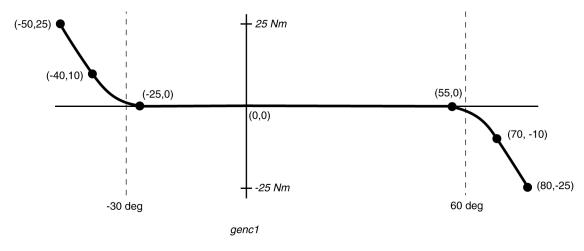


Figure 3-7. Example restraint function.

Since a restraint function is defined for the entire range of motion of a gencoord, it is important to carefully define its values within this range. If you want the function to be exactly zero for a portion of the range (which is usually the case for dynamic simulations), you should define knots (coincident control points) in the appropriate places.

For example, suppose that a gencoord named *genc1* has a range of motion of -30 degrees to +60 degrees, and that if it goes beyond either end of its range, you want to apply a torque to drive it back into its range. Furthermore, suppose that you would like to start applying the restraint torque when the gencoord gets within 5 degrees of its limit, to make it less likely that it will go beyond its range. In this case, your restraint function would look something like the one in Figure 2-1. Note that when the gencoord goes below its range, a positive torque will be applied (to increase the gencoord value), and when it goes above its range a negative torque will be applied (to decrease its value). The function definition would look like this:

```
beginnaturalcubicspline f31
(-50.0, 25.0)
(-40.0, 10.0)
(-25.0, 0.0)
(-25.0, 0.0)
(55.0, 0.0)
(55.0, 0.0)
(70.0, -10.0)
(80.0, -25.0)
endfuntion
```

Note the coincident control points at -25 and 55 degrees. This is very important so that the restraint function has a value of exactly zero between -25 and 55 degrees.

3.3.6 World Objects

World objects are polygonal surfaces that are fixed to the world reference frame. They can be used to enhance the display of a model by providing a setting for the musculoskeletal structure. Typical examples include a floor, walls, stairs, or a bottom bracket and seat for a cycling model. A sample world object is shown in Figure 3-8. The file that contains the object's surface description must be in the same format as a bone file (see Section 3.2, Bone Files), and can be processed with *norm* (see Chapter 4, Norm) before being loaded into SIMM. You can translate, scale, and rotate the object in the bone file using the origin, scale, and rotation parameters, as shown in the figure. The origin defines the location of the origin of the world object in the world reference frame. Rotation defines the XYZ Euler rotation between the world reference frame and the reference frame of the world object (which is implicitly defined by the vertices in the bone file).

To control the appearance of a world object, assign a material to it in the same way that you do for a body segment. If

you do not assign a material, the default material, def_bone, will be used. The drawmode parameter gives you additional control over the shading of the object's surface. As with body segments, flat_shading, gouraud_shading, wireframe, outlined, solid_fill, and none are supported. The default drawmode is solid_fill.

You can also define a texture to be used when displaying the object. The texture map must be an image file in the TIFF format, and you must also create a texture coordinates file, which specifies the UV texture coordinates of each vertex in the bone file. The texture coordinates for a simple world object such as a rectangular floor would look like this:

```
0.0 0.0
1.0 0.0
1.0 1.0
0.0 1.0
```

This means that the first vertex of the floor object is mapped to the upper left corner of the texture image (0.0, 0.0), the second vertex is mapped to the upper right corner of the texture (1.0, 0.0), the third vertex is mapped to the bottom right

```
/* WORLD OBJECT: floor */
beginworldobject floor
                              /* name of this world object */
                              /* bone file describing floor surface */
filename floor.asc
origin 0.0 -0.9 0.0
                              /* XYZ vector from world frame to origin of object */
scale 0.5 0.5 0.5
                              /* scale floor.asc to half size */
rotation 90.0 0.0 0.0
                              /* XYZ Euler rotation of object, in degrees */
                              /* use this material, defined in joint file */
material floor material
                              /* override default and use flat shading */
drawmode flat shading
texture file carpeting.tif
                              /* TIFF images to be used as texture map */
texture coords file floor.tex /* file with UV coordinates for each vertex */
endworldobject
```

Figure 3-8. Example world object

corner, and the fourth vertex is mapped to the lower left corner. For more complicated world objects you may need to use a third-party software package to generate UV coordinates for each vertex. The Wikipedia page on "UV mapping" has some additional information.

Note: you can create a scene, or room, to surround your musculoskeletal model using a set of flat panels with a texture map on each one. An example is provided in the *Models/Texture* folder in your SIMM installation. The model has a floor and three walls. Each object is a 4-vertex bone file with a different TIFF image for the texture map.

3.3.7 Materials

Materials define the color and surface reflectivity of objects. Once you have defined a material, you can assign it to any number of body segments, world objects, muscles, and motion objects. If you do not assign a material to a bone or world object, the default bone material, def bone, is assigned to it. See Figure 3-10 for a complete list of the default materials and how they are used. A sample material definition is shown in Figure 3-9. You should consult the OpenGL Programming Guide for a complete description of material properties, but a brief summary follows. The keywords ambient, diffuse, specular, and emission should each be followed by the red, green, and blue color components of that material property. These values range from 0.0 to 1.0. Ambient refers to the background illumination which hits all sides of an object. Diffuse illumination is brightest where the light source strikes the object perpendicular to the surface. Specular lighting accounts for highlights or glare when the light reflects off the surface directly towards the viewer. Emission is the property of giving off light, which happens equally at all locations on the object's

Figure 3-9. Example material

surface. Finally, the shininess value, which ranges from 0.0 to 128.0, specifies the shininess of the object's surface, and controls the size and brightness of the specular highlight.

Note: When an object (body segment or world object) is displayed in wireframe or solid_fill modes, the color used to draw it is the ambient color specified in the material assigned to that object.

SIMM has eight materials stored internally that it uses to draw various elements of a model. You can override these defaults by re-defining the materials in your joint file. These eight materials, with their default values, are shown in Figure 3-10

3.3.8 Cameras

Cameras (formerly known as *model views*), define the position and orientation, or pose, of a camera relative to the model. Once you have loaded a model into SIMM, you can set the camera pose to any of the poses that you specified in the joints file, by using the **restore camera** command (see

```
beginmaterial def motion /* used for motion objects */
ambient 0.1 0.7 0.1
diffuse 0.2 0.3 0.2
specular 0.2 0.0 0.2
shininess 1.0
endmaterial
beginmaterial def bone /* used for bones and world objects */
ambient 0.7 0.70 \overline{0}.7
diffuse 0.6 0.45 0.4
specular 0.7 0.55 0.4
shininess 10.0
endmaterial
begin material def sel poly /* used for the selected polygon */
ambient 0.2 0.70 0.2
diffuse 0.1 0.45 0.1
specular 0.4 0.65 0.4
shininess 10.0
endmaterial
beginmaterial def bone outline /* used for outlined polygons drawmode */
ambient 0.2 0.0 0.0
diffuse 0.2 0.0 0.0
specular 0.2 0.0 0.0
shininess 90.0
endmaterial
beginmaterial def min muscle /* used for muscles with an activation of 0.0 */
ambient 0.2 0.0 0.0
diffuse 0.4 0.0 0.0
specular 1.0 0.15 0.15
shininess 40.0
endmaterial
beginmaterial def max muscle /* used for muscles with an activation of 1.0 */
ambient 0.4 0.0 0.0
diffuse 0.7 0.0 0.0
specular 1.0 0.15 0.15
shininess 90.0
endmaterial
beginmaterial def muscle point /* used for muscle points when all are displayed */
ambient 0.0 0.0 0.0
diffuse 0.0 0.0 0.7
specular 0.15 0.15 1.0
shininess 90.0
endmaterial
beginmaterial def_sel_muscle_point /* used for selected muscle points */
ambient 0.0 0.0 0.0
diffuse 0.7 0.7 0.0
specular 1.0 1.0 0.2
shininess 90.0
endmaterial
```

Figure 3-10. SIMM's default materials

Section 2.4.2, Command Menu, for details). An example camera definition is shown in Figure 3-11. Each camera is specified by a name and a 4x4 transformation matrix. The 3x3 matrix in the upper left corner specifies the rotations of the model around the axes of the world reference frame. The first three numbers in the last row of the matrix specify the translation in the X, Y, and Z directions from the origin of the world frame (the Z translation should always be negative). You can enter cameras directly into a joint file if you know the exact coordinates, but it is usually easier to manipulate the camera interactively within SIMM, use the **save camera** command (see Section 2.4.2, Command Menu) to save the transformation matrix to a buffer, then write the model to a file so that you have the precise 4x4 matrix saved to a file. You can enter up to five cameras in each joints file.

3.3.9 Wrap Objects

Wrap objects are geometric primitives that automatically deflect muscles to prevent them from penetrating the wrap object. Wrap objects are typically added to a model such that their surface approximates the surface of one or more of the model's bones. The only required component of a wrap object definition is its name. However you will usually want

```
/* Camera Pose */
begincamera posterior /* posterior is the name of this pose */
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
0.2 0.3 -1.0 1.0
endcamera
```

Figure 3-11. Example camera

to specify its associated body segment and its placement within the segment's reference frame as well. Wrap objects may be created interactively within SIMM using the Wrap Editor tool. You may find it easier to create and modify wrap objects with the Wrap Editor rather than editing the joint file directly. See Section 2.9, Wrap Editor, for more information.

```
/* Example Wrap Object */
beginwrapobject hip_wrapper
wraptype ellipsoid
segment pelvis
xyz_body_rotation 20.0 0.0 0.0
translation -0.0826 -0.0286 0.0808
radius 0.05 0.02 0.05
active yes
visible no
quadrant -y
endwrapobject
```

Figure 3-12. Example wrap object

Wrap objects are specified in the model's joint file. However before a muscle can wrap over a wrap object, the muscle must be *associated* with that wrap object. Associations between muscles and wrap objects are specified in the muscle's definition in the muscle file (see Section 3.4.1, Muscles). Muscle wrap associations may also be specified within SIMM using the Wrap Editor tool. The fields in a wrap object definition, shown below, specify the size, position, and orientation of the geometric primitive, as well as method of wrapping.

Wrap Object Fields

wraptype

This parameter specifies the type of geometric primitive to use for the wrap object. The valid types are: sphere, cyl-

inder, ellipsoid, and torus. If no wrap type is specified, the default type, sphere, is used.

segment

This parameter specifies the body segment to which the wrap object is attached.

xyz body rotation

This parameter specifies the orientation of the wrap object as an XYZ Euler rotation. The orientation is calculated by first applying the X rotation around its local X axis, then the Y rotation around its new local Y axis, and finally applying the Z rotation around its new local Z axis. If the rotation is not specified in the wrap object definition, the default values are 0.0, 0.0, 0.0.

translation

This parameter specifies the position of the wrap object's origin with respect to the origin of its body segment. It is expressed as an X, Y, Z translation vector. If no translation is specified, the default values are 0.0, 0.0, 0.0.

object dimensions

The dimensions of the wrap object are specified with several keywords, depending on the type of the object, as described below.

sphere

The radius of the sphere is specified by the keyword radius followed by a single value.

radius 0.15

cylinder

The radius of the cylinder is specified by the keyword radius followed by a single value. The height of the cylinder is specified by the keyword height followed by a single value.

radius 0.10 height 0.25

Note: The cylinder's height is used when calculating muscle wrapping. If *both* of the tangent points (the points of contact between the muscle and the wrap object) are beyond the visible portion of the cylinder, the muscle is considered to miss the wrap object and is not wrapped.

ellipsoid

The sizes of the ellipsoid in the X, Y, and Z directions are specified with the keyword radius followed by the X, Y, and Z sizes, respectively.

radius 0.1 0.2 0.1

torus

The inner and outer radii of the torus are specified by the keyword radius followed by the inner and outer radii, respectively. The outer radius is the distance from the center of the torus to the axis of the curved cylindrical section, and the inner radius is the radius of the cylindrical section itself (and thus should always be less than the outer radius).

radius 0.03 0.08

active

This parameter may be used to disable the wrap object. If its value is no or false, then the wrap object will have no effect on its associated muscles.

visible

This parameter may be used to disable the display of the wrap object's shape in the model window. If this parameter's value is no or false, then the wrap object will not be visible in the model window, but its associated muscles will still wrap over it.

show wrap pts

This parameter may be used to display the XYZ coordinates of muscle tangent points for each muscle that wraps over the wrap object. If this parameter's value is yes or true, then the tangent point coordinates will be displayed in the

model window. The display of these points can be helpful when creating the wrap object or adjusting muscle attachment points to improve the wrapping behavior.

quadrant

This parameter may be used to limit wrapping to one half of the wrap object. The valid quadrants are: x, -x, y, -y, z, or -z for spherical and ellipsoidal wrap objects, and x, -x, y, or -y for cylindrical wrap objects.

wrapmethod

This parameter may be used to specify the algorithm to be used to compute muscle wrapping paths around ellipsoidal wrap objects. The valid wrap methods are: hybrid, midpoint, and axial. For more information, see Section 2.9.5, Ellipsoid Wrapping Methods.

3.3.10 Deform Objects

Deform objects allow you to twist, bend, and warp portions of a body segment that would normally be considered rigid. When a segment is deformed, all bone vertices, distal joint centers, muscle attachment points, and muscle wrapping objects in the deformed region are transformed accordingly. It is usually easier to create new deform objects using the SIMM Deform Editor rather than editing a joint file directly. See Section 2.10, Deform Editor, for information on how to do this. However, it can be useful in certain situations to create or modify deform objects manually by editing the model's joint file.

Deform objects are defined within a segment definition in the model's joint file, within a begindeform ... enddeform pair. All deform object fields have reasonable default values, so it is possible to specify a simple deform with just two lines as shown in Figure 3-13.

```
beginsegment
begindeform my simple deform
enddeform
endsegment
```

Figure 3-13. Minimal deform object specification, located within a segment definition

In most cases, however, you will also define the dimensions of the deform object's inner and outer boxes, the transformation necessary to position the deform within the reference frame of its associated body segment, and the starting and ending transformations of the inner box that creates the object's deformation (see Figure 3-14).

```
deform neck angle
innermin -0.05 -0.05 -0.080
innermax 0.05 0.05 -0.025
outermin -0.07 -0.07 -0.100
outermax 0.07 0.07 -0.005
xyz body rotation POSITION 33.6900 -25.6600 16.1100
translation POSITION -0.0153 -0.0279 0.0287
xyz body rotation DEFORM START -80 0 0
translation DEFORM START 0 0 0
xyz body rotation DEFORM END 80 0 0
translation DEFORM END 0 0 0
enddeform
```

Figure 3-14. Example deform object

If the deform object is an auto-reset deform, then it is not necessary to specify the starting and ending transformations since they will be computed automatically. Also remember that auto-reset deform objects should be the last deform object within a segment. Figure 3-15 shows a typical auto-reset deform object specification.

Deform Object Fields

innermin, innermax

These fields specify the dimensions of the deform object's inner box. All bone vertices, distal joint centers, muscle attachment points, and wrap objects within the deform object's inner box will receive the full effect of the object's deformation. These two keywords must each be followed by three numbers that specify an XYZ coordinate in the deform object's reference frame. The default values for innermin are -0.03, -0.03, -0.03, and the default values for innermax are 0.03, 0.03, 0.03.

outermin, outermax

These fields specify the dimensions of the deform object's outer box. The object's deforming effect is limited to the region enclosed by its outer box. These two keywords must each be followed by three numbers that specify an XYZ coordinate in the deform object's reference frame. The default values for outermin are -0.04, -0.04, -0.04, and the default values for outermax are 0.04, 0.04, 0.04.

```
deform my_auto_reset_deform
autoreset yes
translationonly yes
innermin -0.05 -0.44 -0.08
innermax 0.05 0.05 0.08
outermin -0.07 -0.45 -0.1
outermax 0.07 0.1 0.1
xyz_body_rotation_POSITION 33.6900 -25.6600 16.1100
translation_POSITION -0.0153 -0.0279 0.0287
enddeform
```

Figure 3-15. Example of an auto-reset deform

xyz body rotation POSITION

This field specifies the deform object's rotation relative to its associated body segment. This keyword must be followed by three numbers that specify body-fixed rotations that are applied in X, then Y, then Z order. The default values are 0.0, 0.0, 0.0.

translation POSITION

This field specifies the deform object's translation relative to its associated body segment. This keyword must be followed by three numbers that specify the XYZ translation from the segment's origin. The default values are 0.0, 0.0, 0.0.

xyz body rotation DEFORM START

This field specifies a rotation of the deform object's inner box relative to the deform object's reference frame. This rotation specifies the deform object's starting deformation. The default values are 0.0, 0.0, 0.0.

translation DEFORM START

This field specifies a translation of the deform object's inner box relative to the deform object's reference frame. This translation specifies the deform object's starting deformation. The default values are 0.0, 0.0, 0.0.

xyz body rotation DEFORM END

This field specifies a rotation of the deform object's inner box relative to the deform object's reference frame. This rotation specifies the deform object's ending deformation. The default values are 0.0, 0.0, 0.0.

translation DEFORM END

This field specifies a translation of the deform object's inner box relative to the deform object's reference frame. This translation specifies the deform object's ending deformation. The default values are 0.0, 0.0, 0.0.

active

This field specifies whether the deform object is enabled or not. When disabled, deform objects have no affect on their associated body segment. This keyword must be followed by the word yes, no, true, or false. The default value is yes.

visible

This field specifies whether the deform object is visible in the 3D model window or not. This keyword must be followed by the word yes, no, true, or false. The default value is yes.

autoreset

This field specifies whether the deform object is an autoreset deform or not. If this keyword is followed by the word yes, or true, then the deform is an auto-reset deform and will behave as such. See Section 2.10.5, Combining Multiple Deform Objects, for more information about auto-reset deforms. The default value is no.

translationonly

This field specifies whether an auto-reset deform should limit its deformation to translation only, instead of both translation and rotation. If this keyword is followed by the word yes, or true, and the deform is an auto-reset deform, then the auto-reset deform will translate the segment's origin back to its original, undeformed position but it will not apply a rotation to realign the segment's reference frame with its original axes. The default value is no.

3.3.11 Deformities

Deformity sliders are specified in the model's joint file, after all of the segment definitions. There is no way to create or modify deformity sliders within SIMM, so they must be added to the model's joint file manually. Each deformity defined in the model's joint file appears as a slider at the bottom of the SIMM Deform Editor window.

Deformity Fields

range

This field specifies the minimum and maximum allowable values of the deformity slider. The default values for range are 0.0 to 100.0.

begindeformity anteversion range -20.0 60.0 value 0.0 deform anteversion_twist deform neck_angle deform auto_reset enddeformity

deform

Figure 3-16. Example deformity

This field specifies the default value of the deformity slider.

The default value is 0.0

This field specifies the name of a deform object that has been specified earlier in the same joint file. A deformity slider may contain one or more deform fields. Each deform object associated with the deformity will be controlled by the deformity's slider in the Deform Editor window.

3.3.12 Motion Objects

Motion objects are shapes that are added to the display of the model while a motion is being played back. They are useful for displaying ground-reaction forces, joint contact forces, contact points, and other values calculated by external software packages. The basic shape of the motion object is specified in a SIMM bone file. For each frame of data in a motion file, you can specify the position, orientation, size, and color of the motion object. See Section 3.5.3, Motion Objects, for a description of how to specify these properties.

Motion objects are automatically displayed when any of their parameters appear as columns of data in a motion file. However, before a motion object can be referenced in a motion file it must first be defined in the joint file, or be one of the motion objects built into SIMM. SIMM provides six built-in motion objects named force, ball, contact, joint_force, joint_torque, and spring_force that make it easy to visualize vectors and points in SIMM animations. The ball object is represented by a sphere defined in the SIMM bone file *unit_sphere.asc*. The other five objects are each represented by an arrow defined in the file *arrow.asc*. You can override the parameters of these shapes, or define additional shapes by defining motion object in your joint file.

Motion Object Fields

filename

This field specifies the name of the bone file containing the motion object's shape.

position

This field specifies the starting X, Y, and Z position of the motion object in its parent segment's frame of reference. The default values are 0.0, 0.0, 0.0.

scale

This field specifies the starting X, Y, and Z scale factors of the motion object. The default values are 1.0, 1.0, 1.0.

xyzrotation

This field specifies the starting X, Y, and Z rotations of the motion object (in degrees). The motion object's starting orientation is achieved by first applying the X rotation around its local X axis, then the Y rotation around its new local Y axis, and finally applying the specified Z rotation around its

beginmotionobject box filename unit_box.asc position 0.0 0.0 -0.5 scale 0.1 0.3 0.05 xyzrotation 0.0 45.0 0.0 material my_red_material drawmode flat_shading vectoraxis z endmotionobject

Figure 3-17. Example motion object

new local Z axis (this is known as an XYZ Euler rotation). The default values are 0.0, 0.0, 0.0.

material

This field specifies the name of the material used to determine the color and shading of the motion object's surface. The material must be defined elsewhere in the joint file, or be a default SIMM material. The default material is arrow_mat. See Section 3.3.7, Materials, for information on defining materials, and for a list of default materials.

drawmode

This field specifies the drawing mode for the motion object. The six drawmodes are: solid_fill, wireframe, flat_shading, gouraud_shading, outlined, and none. The default drawmode is gouraud_shading.

vectoraxis

This field specifies the axis of the world object that is used when animating with _vx, _vy, _vz components in the motion file. The motion object will be rotated so that the specified axis coincides with the vector formed by the _vx, _vy, _vz components in the motion file. The motion object will also be scaled along this axis. The default value for this field is y.

3.3.13 Constraint Objects

Constraint objects provide a way of constraining points on one body segment to remain on the surface of a geometric primitive (the constraint object) on a second segment. This mechanism is useful for creating joint kinematics that are too complex to be modeled with a traditional SIMM joint. For example, in the shoulder, two (or more) points on the scapula can be constrained to remain in contact with the ribcage, which is represented as an ellipsoid. Once this constraint is activated, when you change the value of one of the generalized coordinates in the shoulder, SIMM will adjust

the values of the other generalized coordinates in order to satisy the constraints.

It is usually easier to create new constraint objects using the Constraint Editor rather than editing a joint file directly. See Section 2.13, Constraint Editor, for information on how to do this. However, it can be useful in certain situations to create or modify constraint objects manually by editing the model's joint file.

```
/* Example constraint object */
beginconstraintobject r scapula con
constrainttype ellipsoid
segment ribcage
xyz body rotation -9.12 16.21 10.42
translation -0.0399 0.1799 0.0743
radius 0.0800 0.1800 0.0800
active yes
visible no
quadrant -x
beginpoints
/* name
           Х
                    Y
                                   weight
                                            segment tolerance */
    c1 -0.0900 -0.1800 -0.0520
                                   1.0000 r scapula 0.001
    c2 -0.0950 -0.1400 -0.0700
                                   2.0000 r scapula 0.001
   c3 -0.0550 -0.0900
                        -0.0550
                                   1.0000 r scapula 0.002
endpoints
endconstraintobject
```

Figure 3-18. Example constraint object

A constraint object is defined by a geometric primitive (*e.g.*, sphere) fixed to one body segment, and one or more points fixed to other body segments. The points are constrained to remain on the surface of the geometric primitive at all times. The fields in a constraint object definition, shown below,

specify the size, position, and orientation of the geometric primitive, as well as the locations of the points.

Constraint Object Fields

constrainttype

This field specifies the type of the geometric primitive. The available types are: sphere, ellipsoid, cylinder, and plane. If no type is specified, the default type, sphere, is used.

segment

This parameter specifies the body segment to which the geometric primitive (constraint object) is attached.

xyz body rotation

This parameter specifies the orientation of the constraint object as an XYZ Euler rotation. The orientation is calculated by first applying the X rotation around its local X axis, then the Y rotation around its new local Y axis, and finally applying the Z rotation around its new local Z axis. If the rotation is not specified in the constraint object definition, the default values are 0.0, 0.0, 0.0.

translation

This parameter specifies the position of the constraint object's origin with respect to the origin of its body segment. It is expressed as an X, Y, Z translation vector. The default values are 0.0, 0.0, 0.0.

object dimensions

The dimensions of the constraint object are specified with several keywords, depending on the type of the object, as described below

sphere

The radius of the sphere is specified by the keyword radius followed by a single value.

radius 0.15

cylinder

The radius of the cylinder is specified by the keyword radius followed by a single value. The height of the cylinder is specified by the keyword height followed by a single value.

```
radius 0.10 height 0.25
```

Note: the height of the cylinder is used for enforcing the constraint. The constraint points are required to remain on the section of the cylinder defined by the height value.

plane

The width (X dimension) and the length (Y dimension) of the plane are specified by the keyword radius followed by the width and length, respectively.

```
radius 0.2 0.3
```

Note: the width and length of the plane are used for enforcing the constraint. The constraint points are required to remain on the section of the plane defined by these two values.

ellipsoid

The sizes of the ellipsoid in the X, Y, and Z directions are specified with the keyword radius followed by the X, Y, and Z sizes, respectively.

```
radius 0.2 0.3 0.2
```

active

This parameter may be used to disable the constraint object. If its value is no or false, then the constraint object will have no effect on the model.

visible

This parameter may be used to disable the display of the constraint object's shape in the model window. If this

parameter's value is no or false, then the constraint object will not be visible in the model window, but it will still be enforced.

quadrant

This parameter may be used to limit the constraint object to one half of the geometric primitive. When the object is limited in this way, the constraint points are required to remain in contact with the selected half of the object. The valid quadrants are: x, -x, y, -y, z, or -z for spherical and ellipsoidal constraint objects, and x, -x, y, or -y for cylindrical constraint objects.

constraint points

Constraint points are specified between the keywords beginpoints and endpoints. You can define any number of points, attached to any body segments. However, the more points you define the more difficult it will be for SIMM to find a set of gencoord values which satisfy the constraint. In practice, two or three points is usually sufficient to create the desired constraint on the motion.

Each point is defined by its name, its X, Y, Z position in the body segment's reference frame, its weight, the name of the body segment it is attached to, and its tolerance value. The weight of the point is a non-negative floating point value which controls how much emphasis is placed on it when enforcing the constraints. The higher the weight, the more SIMM tries to position that point onto the object surface, at the expense of the other points. The tolerance of a point is the distance (in model units) that it is allowed to be from its correpsonding object, and still be an acceptable solution. It does not affect SIMM's calculation of the solution (as does the weight of a point). Rather, it is a threshold value which is checked after the best constraint solution has been calculated. If the point is not within tolerance of its constraint object, it is colored red to indicate this condition.

3.3.14 Model Poses

A model pose is information about the position and orientation of the model (see Example in Figure 3-19). It can contain the values, velocities, clamped states, and locked states of the gencoords, as well as the activations and display states (visible/invisible) of the muscles. You can define any number of poses for a model, and switch between them in SIMM using the **restore pose** button in the Model Viewer. If you do not define a pose named *default* in your JNT file, SIMM will create one when you load the model. When defining a pose, you can choose to include any subset of the gencoords and muscles. However, for each gencoord that you include, you must specify all of the parameters: name, value, velocity, clamped state, and locked state. For each muscle that you include, you must specify: name, activation, visibility state.

It is recommended that you no longer specify default value in a gencoord definition in a JNT file, or activation or visible in a muscle definition in a MSL file. If you do, SIMM will add attempt to add those parameter values to the default pose of the model (whether the default pose is user-defined or SIMM-generated). If they are already defined in the default pose, SIMM will ignore the values specified in the gencoord and muscle definitions. When you save a JNT file, SIMM will output the default pose, but not the default values in the gencoord definitions or the activation and visible settings in the muscle definitions. In order words, if you do not define a default pose in the JNT file, SIMM will build one using values specified in the gencoord and muscle definitions. If you do define a default pose, all conflicting values specified in the gencoord and muscle definitions will be ignored. And when exporting a model, all pose-related values are written to the default pose, and not to the gencoord and muscle definitions.

```
/* Pose representing mid-stance of right foot during gait */
beginpose right stance
gencoord lower torso TX 0.427888 -1.265419 unclamped unlocked
gencoord lower torso TY 0.296529 0.056601 unclamped unlocked
gencoord lower torso TZ 0.978603 0.147046 unclamped unlocked
gencoord lower torso RX -95.196316 -34.665062 unclamped unlocked
gencoord lower torso RY -7.692675 45.201578 unclamped unlocked
gencoord lower torso RZ 165.942628 6.438312 unclamped unlocked
gencoord lumbar roll -6.413519 -51.289117 clamped unlocked
gencoord lumbar yaw -6.227815 90.353578 clamped unlocked
gencoord lumbar pitch 16.100601 -13.478036 clamped unlocked
gencoord hip add r -1.303244 -25.054797 clamped unlocked
gencoord hip rot r 5.860699 38.671449 clamped unlocked
gencoord hip flex r 20.731736 -116.748438 clamped unlocked
gencoord knee flex r 18.675811 -53.727468 clamped unlocked
gencoord ankle flex r 3.320019 23.947414 clamped unlocked
gencoord subt angle r -29.011612 0.023848 clamped unlocked
gencoord toe angle r 0.000000 -0.000000 clamped unlocked
gencoord hip add 1 8.522440 50.374307 clamped unlocked
gencoord hip rot 1 9.505861 -21.264489 clamped unlocked
gencoord hip flex 1 29.704103 90.207294 clamped unlocked
gencoord knee flex 1 56.643236 -159.016982 clamped unlocked
gencoord ankle flex 1 -13.763388 84.145542 clamped unlocked
gencoord subt angle 1 -27.571145 -6.870360 clamped unlocked
gencoord toe angle 1 0.000000 -0.000000 clamped unlocked
muscle semimem r 0.088945 on
muscle semiten r 0.088945 on
muscle bi fem lh r 0.000000 on
muscle bi fem sh r 0.000000 off
muscle rectus fem r 0.040599 on
muscle semimem 1 0.350749 on
muscle semiten 1 0.350749 on
muscle bi fem lh 1 0.000000 on
muscle bi fem sh 1 0.000000 off
muscle rectus_fem_1 0.021301 on
endpose
```

Figure 3-19. Example model pose. This pose represents the position of the body during the stance phase of the right foot during gait (due to limited space, only some of the gencoords and muscles are shown). Each gencoord line specifies the name, value, velocity, clamped state, and locked state. Each muscle line specifies the name, activation, and visibility state.

3.4 Muscle Files

Muscle files contain descriptions of muscle-tendon actuators. Each musculoskeletal model you load into SIMM must have all of its muscle-tendon actuators defined in one muscle file. If you do not have any muscle-tendon actuators in your model, then you do not need to create a muscle file. When you load a model, SIMM will first check the joint file to see if you specified the name of the muscle file to load with it (using the muscle_file keyword). If no name is specified, SIMM will read the muscle file that you selected with the file browser, if any.

Within muscle files you can define two types of objects: *muscles* and *muscle surfaces*. *Muscles* are biomechanical elements that have lines of action and that can generate forces and moments. They are described in Section 3.4.1. *Muscle surfaces* are display primitives designed to make musculoskeletal models look more realistic. They are described in Section 3.4.2. Ligaments are modeled as muscles with zero-length fibers, and are described in Section 3.4.3.

3.4.1 Muscles

Each muscle-tendon actuator is defined by specifying its geometry and force-generating properties. The geometry is defined by a series of attachment points connected by line segments. A minimum of two points is required to define the muscle path. There are three types of muscle attachment points, described below. An example muscle, illustrating all three types of points, is shown in Figure 3-22.

The most common type of attachment point is called a *fixed point*. A fixed point is rigidly attached to one of the body

segments, with its location specified by XYZ coordinates that are constants.

The second type of attachment point is called a *via point*. Like a fixed point, a via point is rigidly attached to a body segment, but it is active (included in the path) only when the specified generalized coordinate is within a certain range. Via points can be used to implement simple wrapping of a muscle path over a bone or other constraint that obstructs the path only in some body positions. In order to ensure that the muscle length and moment arm changes smoothly when a via point turns on or off, it is important to carefully specify its position along the existing muscle path.

The third type of attachment point is called a *moving muscle point*. In moving muscle points, each of the X, Y, and Z components of the location can be a function of a generalized coordinate, instead of being a constant. For example, you could define a muscle point in the pectoralis muscle to slide over the ribcage as the humerus elevates.

In addition to using via and moving muscle points to wrap a muscle over a constraint, you can also use muscle wrapping objects for a more automated form of muscle wrapping. Wrap objects are defined in joint files, and can be spheres, ellipsoids, cylinders, or torii. Once one or more wrap objects are linked to a muscle, SIMM will automatically wrap the muscle path over those wrap objects. To link a wrap object to a muscle, put the keyword wrapobject in the muscle definition, followed by the object name (as shown in Figure 3-22). For more information about creating and using wrap objects, see Section 2.9, Wrap Editor, and Section 3.3.9, Wrap Objects.

The isometric force-generating properties of a specific muscle-tendon actuator are derived by scaling a generic, Hillbased model. To scale the generic model, you must supply four parameters and three curves. The four parameters are: peak isometric muscle force (F_{\circ}^{M}) , optimal muscle-fiber length $(\boldsymbol{\ell}_{\circ}^{M})$, pennation angle (α) , and tendon slack length $(\boldsymbol{\ell}_{\circ}^{T})$. The three curves are the active and passive force-length relations of muscle, and the force-length relation of tendon. The active force-length curve of muscle is scaled linearly by muscle activation, which ranges from 0.0 (no activation) to 1.0 (full activation). All of the curves are defined by listing control points that are interpolated by natural cubic splines.

There are also some circumstances under which you can model the force-velocity characteristics of the muscle-tendon actuators. When calculating muscle force during a motion sequence, you have the option of estimating the velocities of the muscle fibers and using this information to scale the isometric force computations. This process works by first finding the velocities of the generalized coordinates (either taken from the motion file or calculated by differentiating the generalized coordinate position data. See Section 3.5, Motion Files, for details). Then, muscle-tendon velocity is calculated, and this value is distributed to the fibers and tendon based on their relative stiffnesses. The more compliant element is assigned more of the velocity.

To use these force-velocity features, you need to specify two additional parameters in the definition of each muscle. The first is max_contraction_velocity, which defines the maximum contraction velocity of the muscle fibers in optimal_fiber_lengths per second (a positive number). The second is the normalized force-velocity function for the muscle. This defines the scaling factor for the isometric force, as a function of the normalized fiber velocity (optimal_fiber_lengths per second divided by max_contraction_velocity). Thus the range of the function's abscissa should be -1.0 to 1.0, and its ordinate should

be 1.0 for a velocity of 0.0. When a muscle contracts, it is defined to have a negative velocity.

If you will be using your musculoskeletal model in dynamic simulations, you need to add a section to the muscle file defining the names of the muscle parameters needed by the dynamic muscle model (consult the Dynamics Pipeline manual for details). An example is shown in Figure 3-20. You then need to specify values for these parameters in the muscle definitions.

Note: Case is significant when entering text into muscle files. All SIMM keywords (e.g., beginmuscle, endmuscle) must be in lower-case letters. You are free, however, to use upper- as well as lower-case letters for naming the muscles, and SIMM will display these names exactly as you type them into the muscle file.

You can define a "default" muscle which contains the parameters that are usually identical for every muscle (*e.g.*, the normalized force-length curves). If a particular muscle definition does not contain a certain parameter or curve, the parameter or curve is inherited from the default muscle. This greatly reduces the amount of information that you have to specify in each muscle definition. A sample default muscle is shown in Figure 3-21.

```
begindynamicparameters
damping
activation1
activation2
mass
timescale
enddynamicparameters
```

Figure 3-20. Example definition of dynamic muscle parameters

A sample lower-limb muscle, rectus femoris, is shown in Figure 3-22. The muscle path is defined by four points (for illustration only; it does not represent a realistic path). The first point is a fixed point attached to the pelvis. The second point is a moving muscle point in the pelvis frame. Its X coordinate is a function (f3) of the gencoord hip_flexion, its Y coordinate is a function (f7) of the gencoord hip_adduction, and its Z coordinate is a constant (0.096). The third point is a via point defined in the femur frame, and is included in the path only when knee_angle is between -120 and -80 degrees. This is done so that the muscle path wraps over the femur when the knee is flexed beyond -80 degrees. The fourth point is a fixed point attached to the patella.

The muscle is a member of two groups, hip_flexion and knee_extension, which means that the muscle name will appear in those two muscle-group menus in the Model Viewer and Plot Maker. The muscle definition also lists the values for the four muscle-tendon parameters (F_0^M , ℓ_0^M , ℓ_0^M , ℓ_0^M , ℓ_0^M , ℓ_0^M , and the curves that specify the force-length relations of the muscle

```
/* must use this name for default muscle */
beginmuscle defaultmuscle
tendon force length curve f17
                                    /* force-length curve for tendon */
passive force length curve f18
                                    /* passive force-length curve for */
active force length curve f19
                                    /* active force-length */
force velocity curve f20
                                    /* force-velocity curve */
max contraction velocity 10.0
muscle model 4
                                    /* force model to use for dynamics */
muscle model opt 10
                                    /* force model for optimization of act. */
activation1 7.667
                                    /* coefficient used in activation dynamics */
activation2 1.46
                                    /* coefficient used in activation dynamics */
                                    /* normalized mass of muscle */
mass 0.00286
                                    /* damping coefficient */
damping 0.1
endmuscle
```

Figure 3-21. Example default muscle

and tendon are not given in this example, they would be inherited from the default muscle.

When defining a muscle, you can optionally specify the activation level in addition to the other muscle-specific parameters. Put the keyword activation in the definition, followed by a number between 0.0 (no activation) and 1.0 (full activation). The thickness and color of the muscles in the model window depend on their activations, with the muscles getting thicker and brighter as activation increases. If you do not specify the activation level when defining a muscle, it will be initialized with a value of 1.0 (regardless of the default muscle). In most cases, this is the value that you will want to start with because the Plot Maker uses the

```
/* Rectus Femoris muscle */
beginmuscle RectusFemoris
beginpoints
-0.029 -0.031 0.096 segment pelvis
f3(hip flexion) f7(hip adduction) 0.096 segment pelvis
0.033 -0.403 0.001 segment femur ranges 1 knee angle (-120, -80)
0.012 0.043 -0.001 segment patella
endpoints
begingroups
hip flexion
knee extension
endgroups
wrapobject distal femur ell midpoint range 2 -1
max force
                      780.0 /* Newtons */
                      0.0840
                                /* meters */
optimal fiber length
tendon slack length
                      0.3460
                                /* meters */
                                /* degrees */
pennation angle
                      5.0
min thickness 0.004
max thickness 0.008
min material muscle min
max material muscle max
endmuscle
```

Figure 3-22. Example muscle definition

current activation level when calculating muscle forces (see Section 2.5, Plot Maker, for more details).

You can also optionally specify the minimum and maximum thicknesses and materials of the muscles. At an activation of 0.0, a muscle will be drawn with material min_material and a thickness of min_thickness. At an activation of 1.0, material max_material will be used and its thickness will be max_thickness. For activations between 0.0 and 1.0, a linear combination of the min and max values for the material and thickness will be used. If you do not specify values for these parameters, SIMM will calculate reasonable defaults.

To indicate that the muscle should wrap over a wrap object defined in the model's joint file, you must specify an association between the muscle and the wrap object. Include the wrapobject keyword in the muscle definition, followed by the name of the wrap object to be wrapped over. See Section 3.3.9, Wrap Objects, for more information about defining wrap objects in your joint file. If the muscle is assigned to wrap over an ellipsoid, then the name of the wrap method to be used may follow the wrap object name. The valid wrap method names are: hybrid, midpoint, and axial. If no wrap method is defined, then the hybrid method is used by default. For more information on wrapping methods see Section 2.9.5, Ellipsoid Wrapping Methods.

You may also specify the section of the muscle path that is allowed to wrap over the wrap object. To do this, put the keyword range after the wrap object name (or after the ellipsoid wrapping method, if specified), followed by the indices of the starting and ending muscle points. Only muscle path segments that are between these start and end points will be checked for intersection with the wrap object. A value of -1 for the start point means to start at the first muscle point, and a value of -1 for the end point corre-

sponds to the last point in the muscle. Note that these muscle point numbers refer to the points as they are listed in the definition of the muscle in the muscle file. If there are via points that are currently inactive when muscle wrapping is computed, they are still counted for the purposes of determining the range of muscle path segments to consider for wrapping. If you do not specify a range of points for the wrap object association, then the entire muscle path will be used.

3.4.2 Muscle Surfaces

Muscle surfaces are objects designed to make the display of your musculoskeletal models more realistic. They do not contain any biomechanical properties such as lines of action or optimal fiber lengths, but rather are hollow shells that represent the surfaces of the muscles. Like muscles, each section of the muscle surface is attached to a body segment, so the surface will change shape as you move the joints.

A muscle surface is a roughly cylindrical shape that is defined by a group of lines that run down the length of the cylinder. Each line is defined much like the series of attachment points of a normal muscle, between the keywords beginpoints and endpoints (you can even specify wrapping points). This group of lines is then put between the keywords beginmusclesurface and endmusclesurface to define the complete surface. The points within each muscle line should be ordered in the same direction (e.g., proximal to distal). The sequence of muscle lines, when viewed from the outside, should be ordered left to right, assuming the points within each line are ordered top to bottom. If the points are ordered bottom to top, then the sequence of lines should proceed right to left. The last line of points in the surface definition should be the same as the first, so that the surface is a complete cylinder (Figure 3-23).

To make the muscle surfaces change color with activation, you can specify their activation levels in a motion file. As with muscles, use the name of the muscle surface as a column label, and enter activations from 0.0 to 1.0. The materials *muscle_mat0* through *muscle_mat20*, specified in the joint file, are used to color the muscle surfaces. For example, *muscle_mat0* is used when the activation is 0.0,

```
beginmusclesurface rf surface /* Surface desc. of Rectus Femoris*/
beginpoints
                                           /* line 1: defines the */
-0.0337 -0.0326 0.0877 segment pelvis
 0.0434 -0.0103 0.0059 segment femur
                                           /* anterior contour of */
 0.0594 -0.3291 -0.0072 segment femur
                                           /* the muscle surface, */
 0.0121 0.0439 -0.0090 segment patella
                                           /* proximal to distal */
endpoints
beginpoints
-0.0364 -0.0335 0.0930 segment pelvis
                                           /* line 2: defines the */
                                           /* medial contour of
 0.0392 -0.0108 0.0139 segment femur
 0.0523 -0.3298 0.0060 segment femur
                                           /* the muscle surface, */
 0.0064 0.0433 0.0016 segment patella
                                           /* proximal to distal */
endpoints
beginpoints
-0.0364 -0.0335 0.0930 segment pelvis
                                           /* line 3: defines the
                                                                   */
 0.0392 -0.0108 0.0139 segment femur
                                           /* posterior contour of */
 0.0523 -0.3298 0.0060 segment femur
                                           /* the muscle surface,
                                           /* proximal to distal
 0.0064 0.0433
                 0.0016 segment patella
endpoints
beginpoints
                                           /* line 4: defines the */
-0.0309 -0.0320 0.0931 segment pelvis
 0.0477 -0.0104 0.0139 segment femur
                                           /* lateral contour of */
 0.0664 -0.3291 0.0060 segment femur
                                           /* the muscle surface, */
 0.0177 0.0439 0.0016 segment patella
                                           /* proximal to distal */
endpoints
beginpoints
-0.0337 -0.0326 0.0877 segment pelvis
                                           /* line 5: same as line 1,*/
 0.0434 -0.0103 0.0059 segment femur
                                           /* needed to close the
                                                                     */
 0.0594 -0.3291 -0.0072 segment femur
                                           /* muscle surface
                                                                     */
 0.0121 0.0439 -0.0090 segment patella
endpoints
endmusclesurface
```

Figure 3-23. Example muscle surface

muscle_mat10 is used for activations of 0.5, and *muscle_mat20* is used for activations of 1.0.

You cannot interactively edit the points that define a muscle surface, so as you create them you should make each line an ordinary muscle so you can move the points using the Muscle Editor. Then write the muscles to a file and copy the attachment points to a muscle surface definition.

3.4.3 Ligaments

Ligaments are modeled in SIMM as muscles that have fibers of zero length. The muscle-tendon unit is thus only a tendon, with no active force properties. To define a ligament, create a muscle definition and specify an optimal_fiber_length of 0.0. This tells SIMM to treat the "muscle" as a tendon-only element, using the tendon force-length curve to calculate its force. By specifying a tendon force-length curve that models the mechanical behavior of ligamentous tissue, the "muscle" will thus behave as a ligament. As with muscles, ligaments can span any number of body segments, and can wrap over wrap objects. Because ligaments are created using a modified muscle definition, they will be listed as muscles in the SIMM graphical interface, but you can use muscle groups to differentiate them from normal muscles.

3.5 Motion Files

Motion files contain lists of joint angles that describe motions of musculoskeletal models. Once loaded into SIMM, motions can be linked to the models already loaded. The motion variable (e.g., gait_cycle) is treated as a generalized coordinate. It can be used to animate the model, and

can serve as the independent variable when making plots. You can also create plots of any column of motion data using the Plot Maker (see section 2.5.4, Command Menu). In addition to specifying joint angles during a motion, you can also specify muscle activation levels. Because the thickness and color of each muscle depends on its activation, you can use a motion file to not only playback the kinematics of an animation, but also to display the relative activity levels of the muscles used to create the motion. An example motion file is shown in Figure 3-24.

In addition to joint angles and muscle activation levels, you can also specify animation data for arbitrary objects to be added to the scene when the motion is played back. These are referred to as *motion objects*. SIMM provides several built-in motion objects for displaying forceplates, forceplate forces, markers, foot prints, contact forces, contact points, and joint forces. You can also add your own custom motion objects, and animate their position, scale, orientation, and color. See Section 3.5.3, Motion Objects, for more information about adding motion objects to an animation.

Columns of data in a motion file can correspond to a generalized coordinate (possibly with a suffix), a muscle (possibly with a suffix), a motion object component, or "other data." Other data can be any motion data that you want. It is not used to animate the model when you play the motion, but can be plotted in the Plot Maker.

If you always want to load a certain motion file, or files, with a musculoskeletal model, you can put the name of the motion file inside the joint file, preceded by the keyword motion_file. You can specify up to 100 motion files in this way. See the beginning of Section 3.3, Joint Files, for more details.

```
/* Walking motion */
name gait cycle
                              /* name of this motion */
datacolumns 6
                              /* 6 columns of data */
                              /* 64 rows of data, 1 for each time step */
datarows 64
                              /* press this key to make the model walk */
keys w key
                             /* first column is used as X axis in plots */
x column 1
units sec
                              /* the units of the X column are seconds */
wrap
                             /* when animating, continuously loop motion */
enforce loops no
                              /* enforce loops when animating? */
                             /* enforce constraints when animating? */
enforce constraints no
show markers yes
                             /* show markers when animating? */
show marker trails yes
                             /* show marker trails when animating? */
                             /* show forces when animating? */
show forces yes
show force trails yes
                             /* show force trails when animating? */
                             /* show force plates when animating? */
show force plates yes
                             /* show foot prints when animating? */
show foot prints yes
cursor 1.00 1.00 0.00
                             /* color of cursor in motion plots */
event 0.111111 rhs
                              /* a heel strike event for the right foot */
                             /* a toe off event for the left foot */
event 0.388333 1to
event 0.697917 lhs
                             /* a heel strike event for the left foot */
event 0.821667 rto
                             /* a toe off event for the right foot */
event color 1.000000 0.000000 1.000000
                                         /* color of events in motion plots */
calc derivatives no
                              /* calculate derivatives of all columns? */
/* force plates are specified by size, location, and orientation */
forceplate 0.46 0.51 0.041 -0.642230 0.251 -0.039 -180.0 0.0 -90.0
forceplate 0.46 0.51 0.041 -0.125860 0.252 -0.040 -180.0 0.0 -90.0
/* foot prints are specified by size, location, and orientation */
rightFootPrint 0.208933 -1.215807 0.120825 0.005 0.0 0.0 -77.857691
leftFootPrint 0.278098 -0.868048 0.500083 0.005 0.0 0.0 -109.167633
movie gait cycle.wmv
endheader
time
       hip flexion knee angle
                                                         semimem
                                 ankle angle
                                                rf
0.008
        17.3492
                     -21.7508
                                               0.25
                                                          0.10
                                   4.2261
0.016
        13.8255
                     -21.2797
                                   3.8606
                                               0.24
                                                          0.10
0.024
      12.3931
                     -20.5520
                                               0.24
                                                          0.12
                                   1.6857
0.032
        10.8606
                     -19.2399
                                               0.23
                                                          0.12
                                   0.2093
```

Figure 3-24. Example motion file

When you load a motion file into SIMM using the *File... Open* command in the menubar, the new motion is linked only to the current model, so you should make the model window the topmost one in SIMM before loading a motion file to link to it. Alternatively, you can drag and drop a motion file onto a model window to link the motion to that model.

Note: Case is significant when entering text into motion files. All SIMM keywords (e.g., datacolumns, datarows) must be in lower-case letters. You are free, however, to use upper- as well as lower-case letters for labeling the columns of data, and SIMM will display these names exactly as you type them into the motion file. Spaces are not allowed within the names of the data columns.

3.5.1 Motion File Header

A header in each motion file names the motion, specifies any events, forceplates, and foot prints associated with the motion, and specifies some parameters that control how the motion data is displayed when the model is animated. Here is a description of each header parameter.

name

The name of the motion. This parameter is optional, but should generally always be specified.

datacolumns

The number of columns of data. This parameter is required.

datarows

The number of rows of data. This parameter is required.

x column

The index of the column that will be used as the X axis when plotting motion data. This will be usually be the "time" column, but can also be a normalization column (e.g., 0.0 to 100%) or any other column in the file. SIMM will automatically calculate the minimum and maximum values of the X column data, to serve as the range of the

motion variable. If you do not specify x_{column} , SIMM will use the column labeled "time" as the X column. If there is no column with this label, the first column as the X column.

units

This optional parameter specifies a units label that will be used when plotting motion data. SIMM does not use it to convert any of the data, it merely adds the label that you specify to the labels of the Y variables in the plots.

keys

This optional parameter lets you specify one or two keys to animate the motion. When you press these keys in the model window, the motion will be animated at the *motion speed* specified in the Model Viewer.

wrap

If this parameter is specified, the playback of the motion will wrap around to the beginning when it gets to the end. The motion will thus loop continuously until you stop it.

enforce loops

This optional parameter specifies whether or not to enforce loop constraints while animating the motion. If the motion file contains complete gencoord data that already satisfy the loop constraints, then you can set this parameter to no so that SIMM does not recalculate them. If your model does not contain closed loops of body segments, this parameter has no effect.

enforce constraints

This optional parameter specifies whether or not to enforce constraint objects while animating the motion. If the motion file contains complete gencoord data that already satisfy the constraint objects, then you can set this parameter to no so that SIMM does not recalculate them. If your model does not contain any constraint objects, this parameter has no effect.

show markers

This optional parameter turns on or off the display of the marker data as the motion is animated. Markers are usually

shown as blue spheres, but you can change their display properties by modifying the default motion object.

show marker trails

This optional parameter turns on or off the display of the marker trails as the motion is animated. A marker trail is a white line showing the trajectory of the marker during the entire motion.

show forces

This optional parameter turns on or off the display of the forces as the motion is animated. This controls the display of forceplate forces, contact forces, and any other external forces in the motion data.

show force trails

This optional parameter turns on or off the display of force trails. A force trail consists of a white line representing the force at each time frame, and a series of line segments connecting the tips of the white force vectors.

show_force_plates

This optional parameter turns on or off the display of the forceplates that are specified in the motion header.

show_foot_prints

This optional parameter turns on or off the display of the foot prints that are specified in the motion header.

cursor

If you put this parameter in the header, followed by an RGB color value (see the example in Figure 3-24), SIMM will display a cursor in all plots of the motion data. As you animate the motion, the cursor will move along the plot data in sync with the display of the model.

event

This parameter specifies a motion event, which will be shown in all plots of the motion data. After the keyword event is the X coordinate (usually the time, in seconds) followed by the name of the event.

event color

This optional parameter specifies the RGB color used to show the events in plots of motion data. Events are shown as vertical lines with their names at the bottom.

calc derivatives

This optional parameter specifies whether or not SIMM will calculate first derivatives of the motion data. This option is most useful if your motion contains gencoord data but you also want to plot gencoord velocities. SIMM will use the specified X column to calculate derivatives, so if the X column is not "time," you will not get meaningful results.

forceplate

This parameter specifies a forceplate object that will be displayed as the motion is animated. The built-in forceplate motion object (a unit cube) is used for the display, and it is scaled, translated, and rotated into the proper position based on the specified values. After the keyword forceplate are the XYZ scale factors, the XYZ position of the forceplate origin in the world reference frame, and the XYZ Euler rotation from the world frame. When the SIMM Motion Module imports a TRB/TRC/C3D file and creates a motion, it calculates the proper forceplate information from the analog data, so you normally do not need to specify these values yourself.

leftFootPrint
rightFootPrint

These parameters specify left and right foot print objects that will be displayed as the motion is animated. The built-in foot print motion objects are used for the display, and they are scaled, translated, and rotated into the proper positions based on the specified values. After the keyword leftFootPrint or rightFootPrint are the [single] scale factor, the XYZ position of the foot print origin (back of the heel) in the world reference frame, and the XYZ Euler rotation from the world frame. When the SIMM Motion Module imports a TRB/TRC/C3D file and creates a motion, it calculates the proper foot print information from the heel strike and toe off events, so you normally do not need to specify these values yourself.

movie

This optional parameter, followed by a filename, specifies the movie to load when the motion is loaded. The movie is assumed to already be synchronized with the motion. See Section 3.9, Movie Files, for a list of supported movie formats and information on how to view the movie in the model window.

3.5.2 Gencoords and Muscles

The most common types of data found in motion files are generalized coordinate values and muscle activation levels. You can also specify velocities of one or more generalized coordinates in a motion file. These velocities can be used to estimate fiber velocities when calculating muscle forces during the motion. To specify the velocity of a gencoord, the column name must be the name of the gencoord appended by "_vel". Note: it is usually preferable to have SIMM calculate gencoord derivatives by specifying the calc_derivatives parameter in the motion file header, rather than defining them explicitly in the motion file.

Since the thickness and color of the muscles in the model window depend on their activation levels, specifying muscle activations in a motion file can create more informative animations. As the body segments are animated through the motion sequence, the muscles will change their appearance to reflect their relative levels of activity. Muscle activation levels must be in the range 0.0 to 1.0. To specify a muscle's activation in a motion file, the column name must be the name of the muscle.

3.5.3 Motion Objects

SIMM allows you to visualize phenomena such as ground reaction forces and joint contact points by animating arbitrary shapes when a motion is played back. By default SIMM provides an arrow for visualizing vectors and a sphere for visualizing points.

Animation data for motion objects are labeled in the motion file by combining the name of the body segment to which the object should be attached with the name of the motion object and the name of the component to be animated. The components that may be animated are:

• position: _px, _py, _pz

• scale: _sx, _sy, _sz

vector: _vx, _vy, _vz

color: _cr, _cg, _cb

For example, you can display force vectors attached to any body segment (e.g., ground) by specifying their point of application and size in a motion file. For each vector, enter six columns of data in the file, three for the X, Y, and Z coordinates of the point of application, and three for the X, Y, and Z coordinates of the vector itself. The column labels should be the name of the body segment to which the vector is attached, appended by _force_px, _force_py, and _force_pz, for the point of application, and _force_vx, force_vy, and _force_vz for the force vector.

The names of the motion objects built into SIMM are force, ball, contact, joint_force, joint_torque, and spring_force. You can override the properties of these six motion objects, or add additional motion objects in the joint file. See Section 3.3.13, Motion Objects, for more information about defining motion objects.

The bone file used to display the built-in force motion object is named *arrow.asc*. This shape is an arrow of unit length with the tail of the arrow at the origin and the head of the arrow pointing up the Y-axis. This produces force vectors that are displayed with their tail positioned at the point of application, and their arrow head pointing away from the point of application. If you prefer to have force vectors dis-

played with the arrow head pointing toward the point of application, you can use norm to translate the shape defined in "arrow.asc" from the positive Y axis to the negative Y axis with the following command: norm arrow.asc arrow.asc -ty max.

Lastly, the built-in force motion object includes XYZ scaling factors designed to produce force vectors of a reasonable size when forces are specified in newtons and your model length units are meters. You can override the default scaling factors by adding a motion object definition to your joint file like this:

beginmotionobject force scale 1.0 0.002 1.0 endmotionobject

Figure 3-25. Overriding built-in motion object parameters. The scale factors shown here are the SIMM defaults for the built-in force motion object.

Forceplate Objects

Forceplates are special motion objects that are defined in the header of the motion file and are not driven by columns of motion data. Their position and color are constant throughout the motion, so each one can be fully specified in the header with XYZ scale factors, an XYZ translation, and an XYZ Euler rotation. The bone file used to display forceplates is *forceplate.asc*, located in the *Resources\bones* folder in your SIMM installation. When the SIMM Motion Module imports a TRB/TRC/C3D file and creates a motion, it calculates the proper forceplate information from the analog data, so you normally do not need to specify these values yourself.

Foot Print Objects

Like forceplates, foot prints are also special motion objects that are defined in the header of the motion file and are not driven by columns of motion data. Their position and color are constant throughout the motion, so each one can be fully specified in the header with a scale factor, an XYZ translation, and an XYZ Euler rotation. The bone files used to disleftFootPrint.asc play foot prints are and rightFootPrint.asc, located in the Resources\bones folder in your SIMM installation. When the SIMM Motion Module imports a TRB/TRC/C3D file and creates a motion, it calculates the proper foot print information from the heel strike and toe off events, so you normally do not need to specify these values yourself.

3.5.4 Other Data

Columns of motion data that do not correspond to generalized coordinates, muscles, or motion objects are called *other data*. These columns must contain datarows elements, like all data columns, but they do not need to correspond directly to anything in a model. For example, they can be used for plotting [previously-computed] joint moments or accelerations during gait. You can put columns of *other data* anywhere in the motion file (just make sure that the data lines up with the column names). When SIMM reads the motion file, it will automatically figure out which columns correspond to model components (e.g., generalized coordinates), and which ones are *other data*. It will print the *other data* column names to the message window so you can see which ones SIMM did not associate with a model component.

3.5.5 Standard Deviation Data

For any of the columns of data in a motion file, except muscle activations and motion object components, you can also specify standard deviations of the values for each time frame. This is useful if you averaged recorded data from several motions and want to display the standard deviations along with the averages. To specify them for a column of data, add a second column of data, containing the non-negative values for one standard deviation, and label the column with the original data label appended by _std. When you plot the data using the **motion curves** command in the Plot Maker, the standard deviations are displayed underneath the data, in the same color.

3.5.6 Auxiliary Data

Auxiliary data is a column of data in a motion file that is linked to another column. It has many uses, but the most common is for storing onset/offset data for muscles, and linking them to the corresponding column of muscle EMG or activation data. For example, when the SIMM Motion Module imports a motion trial with EMG data, it rectifies, smooths, and scales the raw EMG data and stores the result as a column of muscle activation, using the muscle name as the column label. The Motion Module also uses a thresholding algorithm on the processed EMG to calculate onset times, offset times, and amplitudes. The resulting step function is stored in an auxiliary data column linked to the activation column. The label for an auxiliary data column is always the label of the primary column appended by aux. When you plot the data using the **motion curves** command in the Plot Maker, the auxiliary data are displayed on top of the primary data, in a lighter shade of the same color.

3.6 Plot Files

Plot files contain sets of coordinates describing plot curves. They can be loaded into SIMM and added to plots in order to compare with other curves. They are designed to allow you to compare data derived from other sources (*e.g.*, experimental data, dynamic simulations) with the data that SIMM computes from your model. SIMM can also write out plot files containing curves that were computed from your musculoskeletal model. The format used for input and output is the same, so you can create a plot with the Plot Maker, save it to a file, and load it back into SIMM at a later time. Also, because the format is tab-delimited text, you can easily import plot files into other software applications for additional analysis.

To save a plot in SIMM to a file, make sure the plot window is the topmost window, then select *File... Save Plot* from the SIMM menu bar. To load a plot file into SIMM, select File... Open from the menu bar and browse for the file.

An example plot file is shown in Figure 3-26. Comments are indicated by a hash mark (#). All text on a line that follows a # is ignored by SIMM when reading the file. You can specify a title for the plot using the title keyword. All text on the line following title is interpreted as the title of the plot. You can specify plot curves individually, or in a table format (much like the columns of data in a motion file). In the example in Figure 3-26, three curves are specified, two in the first section, and one in the next. After the title, the X and Y variables and curve name are specified for the first set of curves. The name of the X variable is ankle_angle. If you plan to load this curve into an existing plot to compare to other curves, it is important to give the X and Y variable.

```
# Ankle plantarflexion moment
# Sale, D.J., et al., Influence of joint position on
# ankle plantarflexion in humans, J. Appl. Physiol.:
# Resp. Environ, Exercise Physiol. 52(6): 1636-1642, 1982
# pos angles = dorsiflexion
# neg angles = plantarflexion
        ankle angle moment vs. ankle angle
title
                Sale (ave 90) | ankle angle moment
ankle angle
                Sale (knee 90) | ankle angle moment |
-30.0
        -35.0
                -38.0
                        5.5
-25.0
        -50.0
                -40.0
                        5.9
-15.0
        -79.0
                -63.0
                        7.0
 -5.0 -114.0
                -87.0
                        6.8
  5.0 -144.0 -125.0
                        6.2
 15.0 -170.0
             -150.0
                        7.1
 20.0 -168.0
               -151.0
                        7.0
ankle angle
                Sale (knee ext) | ankle angle moment
-30.0
        -52.0
-25.0
        -75.0
-20.0
       -87.0
-15.0 -105.0
-10.0 -125.0
 -5.0 -140.0
  0.0 -155.0
  5.0 -165.0
 10.0 -170.0
```

Figure 3-26. Example plot file

ables names that exactly match the names used in the other curves. Next comes the text describing each plot curve. Each curve's text should be preceded by a tab. The format for this text is: curve name followed by a vertical bar (|) followed by the name of the Y variable. You can also include standard deviation data for the Y variable by entering a second vertical bar followed by the keyword std. The name of the first curve in the example is Sale (ave 90) and

its Y variable is ankle_angle moment. The name of the second curve is Sale (knee 90) and its Y variable is ankle_angle_moment. Standard deviation data is supplied for the second curve. After specifying all of the curve names and Y variables (for the first set of curves), you then enter the columns of plot data. The first column contains the values of the X variable. The second column contains the data for the first curve, and the third and fourth columns contain the data and standard deviations, respectively, for the second curve.

The second set of plot curves in the example file contains just one curve. This curve has the same X and Y variables as the other two curves, but it is defined separately because it uses a different range of the X variable ankle angle.

3.7 Model Archive Files

A model archive is a compressed file containing all of the files associated with a model, including joint file, muscle file, bones files, and any motion files currently linked to the model. It contains everything needed to load the model back into SIMM at a later time. To create a model archive, select *File... Export Model Archive* from the SIMM menu bar. To open an archive, select *File... Import Model Archive* from the menu bar. Model archive files cannot be opened in other software applications. However, because everything is in one file, it is a good way to share models with other users, or to backup your models before modifying bone files or other components that are typically shared among models.

3.8 Script Files

A script file is a text file of commands that can be executed by the Scripting Tool. Section 2.19, Scripting Tool, lists all of the available commands. An example is shown in Figure 3-27. You can create a script file in any text editor, and load it into SIMM in one of three ways. You can select *File... Open* in the menu bar and choose the script file; you can type "load <filename>" into the Scripting Tool; and you can drag and drop the file onto any SIMM window. For drag and drop, the window on which you drop the file affects how the commands are executed, as explained in Section 2.2.1, Drag and Drop.

```
cd D:\GaitAnalysis\MotionData\subject01
pref MOCAP MODEL D:\GaitAnalysis\Model\GaitModel.jnt
open mocap static.trb
open walk1.trb walk2.trb
tool pm
yvar norm musculotendon length
xvar walk1
muscles semimem r on
muscles semiten r on
muscles semimem 1 on
muscles semiten 1 on
make
xvar walk2
make
open D:\GaitAnalysis\NormalData\NormalHamstrings.plt
rename "norm musculotendon length vs. walk1" "MyHamstringsPlot"
save plt hamstrings.plt MyHamstringsPlot
close "Gait Model"
close MyHamstringsPlot
```

Figure 3-27. Example script file. In this example, the mocap model Gait Model is loaded, using static.trb as a static trial. Then two walking motions are loaded, walk1.trb and walk2.trb. A plot of hamstrings lengths is created, plotting left and right semimem and semiten as a function of each walking motion. An external plot file is then loaded and its data is added to the plot. The plot is then renamed and saved to a file. Lastly, the model and plot are closed.

SIMM uses a script file to save and restore settings each the program is launched closed. If the or WRITE STARTUP SCRIPT preference is on, SIMM will write a script each time the program is closed. The file is named SIMM.ssc and is written to your user application data folder (e.g., C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\6.0\SIMM.ssc). If the READ STARTUP SCRIPT preference is on, this file will be read by SIMM the next time the program is launched. The file contains commands to open, close, and position tool windows so that the set up of the windows is preserved from one use of SIMM to the next. You can customize this file by adding any commands you but do be sure want, vou SO. WRITE STARTUP SCRIPT off so that your modifications are not overwritten when you exit SIMM.

In addition to the session-based script file, SIMM can also use model-based scripts to save and restore settings for individual models. If the write script file parameter is set to on in a joint file, SIMM will write a script file each time the model is saved or closed. The file is created in the same folder as the joint file, and has the same base name with the suffix ".ssc". The file contains commands to restore the settings in the Model Viewer, Plot Maker, Motion Reporter, Motion Editor, and Dynamics Tool to the same values as when the file was created. If the read script file parameter is set to on in the joint file, SIMM will read this script file each time it loads the model, and use it to restore the settings in those tools. You can customize this file by adding any commands you want, but if you do so be sure to turn off the write script file parameter so that your modification are not overwritten when you save or close the model.

Note: model archives contain their corresponding script files within the archive. If there is a separate script file in the same folder as the archive, it will not be read when the model is loaded.

3.9 Movie Files

Movie files can be loaded into SIMM and displayed in the model window in conjunction with motions. There are several ways to load a movie into a SIMM model, but in all cases the movie is linked with one or more specific motions in that model. The movie file formats that SIMM supports are WMV, AVI, MPG, MOV, and Mp4. Here are the three ways of loading a movie file into SIMM.

auto-load with motion trial

When you load a TRB/TRC/C3D/IKS file (motion trial), SIMM checks to see if there is a movie file in the same folder with the same base name. If there is, the *load movie* checkbox in the dialog box is enabled and checked. If this checkbox is checked when the motion trial is loaded, SIMM will automatically load the movie and link it to the motion.

load with motion file

In the header of a motion file, you can specify the name of a movie file to load with the motion:

movie <name of movie file>

When the motion file is read, the specified movie file will be loaded and linked to the motion.

drag and drop

When you drag and drop a movie file onto a model window, SIMM does one of the following:

* If no motions are in the model, SIMM will display an error message.

- * If one motion is in the model, SIMM will link the movie to that motion (replacing the previous movie, if any).
- * If more than one motion is in the model, SIMM will display a dialog box showing all of the motions (and any movies linked to them). You can use the checkboxes to select one or more motions to which to link the movie.

To see a movie during motion playback, there must be a bone named *moviescreen.obj* in the model (there can be more than one). A bone with this name is in the *SIMM\Resources\bones* folder. You can drag and drop it onto a model window to add a screen to that model. This can be done before or after loading a motion with a movie. To reposition the screen, right-click on it and choose *edit moviescreen.obj bone*. You can then use the Bone Editor commands to translate/rotate/scale the screen.

Note: When a movie is linked to a motion, SIMM assumes that the entire movie is time-synched with the entire motion (e.g., the first video frame is displayed with the first motion frame, and last video frame is displayed with the last motion frame). When a motion is cropped in the Motion Editor, the linked movie is cropped as well.

Note: When a movie is loaded into SIMM, its frames are stored in an uncompressed format for faster playback. If your computer does not have a lot of memory, you should probably not load more than 1 or 2 short movies at a time.

4 Norm

Norm is a utility program that you can use to preprocess your bone and world object files (collectively called "object files") before you can load them into SIMM. The main purpose of *norm* is to compute surface normals for the polyhedra in the files, but it can also perform other functions, such as removing duplicate vertices, making the polygons convex, and translating or rotating the polyhedra. These features can be useful if you are creating new object files from medical image data or 3D digitization and you want to load them into SIMM.

Most of the functionality of *norm* is accessible via the Bone Editor tool in SIMM. This graphical interface lets you view desired rotations, translations, and scales to a bone before they are applied with *norm*. The Bone Editor also has simple polygon and vertex editing functions, which *norm* does not. However, the stand-alone version of *norm* contains some file conversion and advanced normal calculation functions that are not accessible in the Bone Editor. Before you begin modifying your bone files, it is recommended that you learn about the Bone Editor and see if it can perform all the functions you will need. If not, then use the stand-alone version of *norm* described below.

Norm is able to process object files which contain more than one object (*e.g.*, a single input file containing all of the ribs of an upper-body model). After reading in the file, *norm* will separate out the individual polyhedra, compute surface normals for each of them, then write them all to the

specified output file. The polyhedra in an object file do not have to be closed surfaces (they can have holes in them). However, *norm* may have difficulty computing surface normals for such polyhedra. If *norm* reports that it was unable to compute normals for one or more polyhedra, it will still create an output file, but some of the surface normals will be incorrect. These polyhedra will not be shaded properly in SIMM, but you can still use them as object files in your model. Also, if there are concave polygons in the object file, *norm* may have difficulty computing surface normals for the object. However, *norm* has an option (described below) to subdivide polygons as it processes the file.

To process an object file, open a DOS window, change directories to where *norm* is installed (it is in the same directory as the SIMM executable) and type norm followed by one or more file names and the desired options, if any. *Norm* can read any of the three file formats described in Section 4.2, Bone Files, and by default writes out bones in the newer ASCII format. File names may appear anywhere on the command line. In other words, you can intersperse options and file names in any order. *Norm* identifies file names by their lack of an initial dash (-). Here are some examples of how you can specify file names, and the rules that *norm* follows when processing them:

norm file

Norm will process file, and save the output object in file. It will prompt you to confirm that the contents of file will be overwritten.

norm infile outfile

Norm will process infile and save the output object in outfile.

norm infile1 outfile1 infile2 outfile2 ...

Norm will process infile1, infile2, ... and save the output objects in outfile1, outfile2, ... respectively.

norm -files *.asc

All files in the current directory ending with .asc will be passed to *norm* as input files. For each file, *norm* will process it and then prompt you to make sure you want to overwrite the file with norm's output. Use of the -files option may be combined with the use of input/output file name pairs on the command line.

Norm has several options that allow you to perform simple geometric transformations on your object files. They can be helpful when creating a musculoskeletal model from raw bone-surface data. Below is a complete list of *norm*'s options:

-overwrite

automatically overwrite existing files without prompting.

-outdir dirname

put all output files in the specified directory. If the directory doesn't already exist, norm will create it.

-files *filter*

process all files in the current directory that match the specified filter. These files will be overwritten with the output.

-sep

write out each polyhedron that is in the input file to a separate output file. The name of each output file will be based on the single output file name you specify (for each input file).

-tol *num*

this option sets the tolerance that *norm* uses when checking for duplicate vertices. All vertices that are within *num* units of each other will be merged into one vertex. A value of 0.0 (which is the default) stops *norm* from checking for duplicate vertices.

-fill fill in holes in the polyhedra. As *norm* processes a polyhedron, it will find holes in it and attempt to fill them in with

triangles. If the polyhedron contains large holes, *norm* may have difficulty filling them.

-es num

this option lets you specify the maximum allowable length for edges in the polyhedra. If *norm* finds any edges with a length greater than *num*, it will subdivide them until they are shorter than this length. *Norm* does not create additional polygons during this process; it just inserts vertices into existing edges.

- -con make all of the polygons in the file convex. *Norm* will split off pieces of concave polygons (making new triangles) until all of the polygons are convex.
 - -tri triangulate all of the polygons in the file. *Norm* will divide all polygons with four or more edges into triangles.
- -ccw tells *norm* that all polygons are ordered in a counter-clock-wise direction (i.e., the vertices within each polygon are specified counter-clockwise). If you use this option, *norm* will not reorder any polygons, but will calculate vertex normals assuming that all of the polygons are counter-clockwise.
- -cw tells *norm* that all polygons are ordered in a clockwise direction (i.e., the vertices within each polygon are specified clockwise). If you use this option, *norm* will reorder every polygon so that they are counter-clockwise, and will then calculate vertex normals.
- -rn *opt* this option is used to specify a reference normal which *norm* will use as a last resort when calculating vertex normals. *Norm* will attempt to determine the inside and outside of a polyhedron without using a reference normal. If the polyhedron is not a closed object, however, it may have difficulty finding the outside. If *norm* does have difficulty, and you do not know if the polygons are all clockwise or all counter-

clockwise, you should specify a reference normal. This vector corresponds to the major direction of the surface normal of the topmost polygon of the polyhedron. For *opt*, use one of: X, -X, Y, -Y, Z, -Z. The default value of the reference normal is Y.

-oldascii

write the polyhedron to an old-style ASCII file. This style of ASCII was the required input format for early versions of *norm*, but has since been replaced with the new ASCII format. Use this option if you want to output a polyhedron without bounding cube information or vertex normals.

-binary

write the polyhedron to a binary file. This option is provided so that you can create polyhedron files in the format which early versions of SIMM required, but it is recommended that you create ASCII files, and not use this option.

-tx *num* add *num* units to all vertex X coordinates.

-ty *num* add *num* units to all vertex Y coordinates.

-tz *num* add *num* units to all vertex Z coordinates.

-rx *num* rotate all vertex coordinates *num* degrees around the X axis.

-ry *num* rotate all vertex coordinates *num* degrees around the Y axis.

-rz *num* rotate all vertex coordinates *num* degrees around the Z axis.

-sx *num* scale all vertex X coordinates by a factor of *num*.

-sy *num* scale all vertex Y coordinates by a factor of *num*.

-sz *num* scale all vertex Z coordinates by a factor of *num*.

For the tx, ty, and tz options, you can specify certain keywords instead of entering an actual number. These are explained below. Shown are the options only for tx, but the options are the same for ty and tz:

-tx min translate the origin of the object along the X axis so that the smallest X coordinate is 0.0 (i.e., find the smallest X coordinate among all the vertices and subtract it from all the X coordinates).

-tx max translate the origin of the object along the X axis so that the largest X coordinate is 0.0 (i.e., find the largest X coordinate among all the vertices and subtract it from all the X coordinates).

-tx mid translate the origin of the object along the X axis to the midpoint between the smallest and largest X coordinates (i.e., average the smallest and largest X coordinates and subtract this value from all X coordinates).

The geometric operations are all implemented as spacefixed transformations, and are performed in the order in which they are entered on the command line (from left to right).

5 **Motion Module**

5.1 Introduction

The Motion Module is an optional component to SIMM that allows you to easily import data recorded by a motion capture system. It reads files containing tracked marker data (3D positions of markers in global space) using the TRC or TRB file format developed by Motion Analysis Corporation. It can also read analog files in the ANB or ANC format with ground-reaction force and EMG data that was recorded in sync with the motion. The Motion Module can also read C3D files, which contain both tracked marker and analog data in the same file. Additionally, the real-time version of the Motion Module can connect to a Motion Analysis system and receive and display motion and analog data in real-time, as it is being recorded.

Files of tracked marker data contain a sequence of frames, each representing a snapshot of the subject's motion at a particular instant in time. Each frame contains the X, Y, and Z coordinates, expressed in a global coordinate system, of all the identified markers. A frame of marker data can thus be thought of as a "marker cloud" because the coordinates are not organized by body segment.

The Motion Module imports tracked marker data and fits a SIMM model within the marker cloud for each time frame. If the SIMM model contains markers whose names and positions match those of the markers placed on the subject, the Motion Module can adjust the model's gencoord values

to determine a "best fit" of the model to the marker cloud. The quality of a fit is determined by how closely each of the model's markers is to its corresponding marker in the marker cloud. It then uses this best fit as the starting position for solving the next frame of data. The result is a SIMM motion that matches the tracked marker data. The model that is used to fit the data can either be one that you create or the pre-made model (the *mocap model*) that comes with the Motion Module.

The Motion Module has two primary components. The first component reads files containing tracked marker data (in the TRC, TRB, or C3D format) and creates SIMM motions from them, as described above. For more information on how this process works, and the various options for importing marker data, see Section 5.2, Opening Tracked Marker Files.

The second component of the Motion Module creates a musculoskeletal model of a given individual by scaling a generic full-body model (the *mocap model*) based on tracked marker data from a static pose. The algorithms that are used to scale the model are the same as those used in OrthoTrak, a full-body gait analysis package available from Motion Analysis. For more information on the mocap model and how it is created and used, see Section 5.3, Using the Mocap Model.

5.2 Opening Tracked Marker Files

SIMM can import tracked marker data that is stored in either a TRB or TRC data file. These file formats, described in the Cortex manual, contain X, Y, and Z coordinates for each identified marker for each time frame. You can also import analog data files containing forceplate and EMG

data recorded during the motion. These analog data files can be in either the ANB or ANC formats. The Motion Module can also read XLS files containing other motion-related data that you may want to view in SIMM, such the kinetic data contained in an OrthoTrak single trial spreadsheet. For more information on importing analog and XLS files, see Section 5.2.3, Analog Data.

The Motion Module can also read C3D data files. These files contain tracked marker and analog data in the same file, so you only need to load one file to import all of your motion data from a trial.

When you open a tracked marker file (along with any associated analog files), SIMM attempts to map the data onto the current musculoskeletal model, thus creating a SIMM motion that is linked to the model. Therefore, to open a tracked marker file, you must already have loaded into SIMM a model that contains the same marker set used in the marker file. For best results, you should make sure that every marker in the tracked marker file is also in the SIMM model, and that their locations in the SIMM model match where they were placed on the subject. The marker names should match exactly (except that they are case-insensitive). If the file contains markers that are not in the model, their data will be ignored by the Motion Module. Similarly, if the model contains markers that are not in the file, they will not be used to help fit the model to the motion data.

If you need to add, rename, or move markers in your SIMM model before loading a tracked marker file, you can use the Marker Editor to do so. See Section 2.12, Marker Editor, for more details.

5.2.1 Selecting Tracked Marker Files

To import a tracked marker file into SIMM, you can simply drag the file from a Windows file explorer and drop it onto the model window you want to import it to. You can drag and drop several files at once and SIMM will import them sequentially. You can also load tracked marker files using *File... Open* in the menu bar, after first making sure that the model you want to import them to is the current model. After dropping the files (or selecting *Open* in the file browser), SIMM will display a dialog box with several options for importing each data file into SIMM. After dismissing the dialog, you can cancel the import of a tracked marker file while its frames are being IK solved by pressing the *Esc* key at any time.

Note: If your analog data files have the same base name as your TRB/TRC file (e.g., subject14.trc, subject14.anb, and subject14.xls), then it is not necessary to drag and drop analog files or select them in the file browser. SIMM will automatically open any analog or XLS files with the same base name and in the same folder as the tracked marker file (there is an option in the dialog box to turn off this feature). If you are loading C3D files, this is not an issue since all of the data for the motion are stored in the C3D file.

5.2.2 Tracked Marker Options Dialog

Once you have selected one or more tracked marker files using the process described in the previous section, SIMM displays a dialog box for each one (in sequence), allowing you to set some options for importing the marker data. In many cases, you will want to use the default settings for these options, so you can simply click the OK button to import the motion. The following list describes each option in the dialog box.

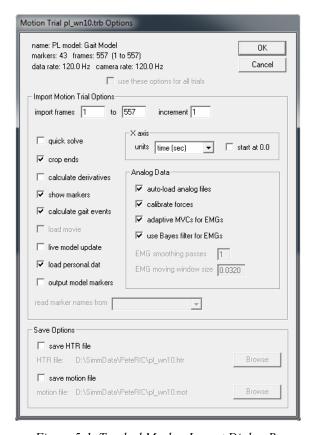


Figure 5-1. Tracked Marker Import Dialog Box

use these options for all trials

This checkbox is enabled when you select more than one tracked marker file for import. If checked, SIMM will use the current settings in the dialog for importing all remaining files, instead of presenting the dialog when each file is opened. If you keep the first and last frame fields (see the next option below) at their initial values, SIMM will assume that you want to import all frames of all files. If you set them to any other values, SIMM will use those same values on all files. For example, if you set them to 100 and 200, SIMM will import only frames 100 to 200 of each file. If this checkbox is checked and you also choose to save HTR

or motion files, SIMM will generate appropriate names for each output file and write it to the folder chosen in the dialog. Note: if this box is checked and you press the *Cancel* button, the import of all remaining files will be cancelled.

Import Motion Trial Options

import frames, to, increment

These fields allow you to specify the range of frames to read from the marker file, as well as the increment. To use them, type into the first two fields the starting and ending frame numbers that you want to import. The third field specifies the increment to use when reading frames from the file. For example, to read every other frame from the file, enter an increment of 2. The starting frame number field and the increment field are initialized to 1. The ending frame number is initialized to the number of frames in the marker file.

quick solve

The Motion Module contains two optimization algorithms for fitting the musculoskeletal model to the marker data. The default method is fairly robust- it is designed to handle cases in which several markers are missing from a frame or in which the markers move large amounts between frames. The other method, called *quick solve*, is less robust but works up to twice as fast as the default method. If speed is an issue, and you know that your marker data is well-behaved, you may want to turn this option on to use the faster optimization algorithm.

crop ends

Tracked marker data files often have frames at the beginning and end of a motion that are missing some markers (because the subject is outside the camera volume). To automatically detect and ignore these frames as the file is read, turn on this option (it is on by default). When the option is on, SIMM will start at the first frame and delete it if it is missing one or more markers. It will then continue to scan forward through the frames, deleting each one, until it encounters a frame containing all of the markers. It will then

do the same procedure starting at the last frame and working backwards. SIMM will not remove frames with missing markers that are in between full frames, so there may still be frames in the motion that are missing markers.

calculate derivatives

When loading a motion, SIMM has the capability of calculating derivatives of the motion variables. When this option is turned on, after SIMM has solved the marker data and created a SIMM motion, it will calculate first-order time derivatives of the generalized coordinate values (*i.e.*, joint velocities) during the motion. It will also calculate derivatives of any force or EMG data in the analog file (if present). These derivatives can then be plotted using the **motion curves** command in the Plot Maker (see Section 2.5.2 for more details). To calculate derivatives, SIMM uses the time column in the motion data and assumes that its units are seconds. If there is no time column or if it is not in seconds, the derivatives that SIMM calculates will not be correct.

show markers

This checkbox turns on the display of the global marker positions in each frame when playing back a motion. When it is on, SIMM will add spherical motion objects to the motion, representing the location of each marker, as recorded in the marker file. When you animate the model according to the motion, the blue spheres represent these actual, recorded marker locations. These are the marker locations that the Motion Module is trying to fit the model to, for each frame. It can be helpful to display them in the model window in order to visualize how good the fit is, and to help debug problems with the data.

calculate gait events

This checkbox turns on the calculation of gait events. Heel strike and toe off events are calculated using a combination of force plate forces and trajectories of the sacral, heel, and toe markers. Essentially, the type of event (right vs. left, heel strike vs. toe off) is determined by the marker trajectories, and the timing of the event is determined by the force

plate forces. If no forces are available for the event, the timing is calculated from the marker trajectories.

load movie

This checkbox controls the import of a movie file associated with the motion trial. When a motion trial is selected for import, SIMM checks to see if there is a movie file in the same folder with the same base name. If there is, this checkbox is enabled and checked. To see the movie in the model window, synchronized to the motion, there must be an object called *moviescreen.obj* in the model. See Section 3.11, Movie Files, for more information about importing and displaying movies.

live model update

When this option is turned on, the model window will update with each frame of IK data as it is calculated. However, all SIMM commands (e.g., to change the model view) are still unavailable until the motion import is complete.

load personal.dat

When this option is turned on, SIMM will read *personal.dat* with each motion trial, and update the subject information in the Motion Reporter accordingly.

output model markers

When this option is turned on, SIMM will write to the generated motion the positions (in the ground frame) of all of the model's markers. These are not the positions of the recorded markers in the input motion. Instead, after SIMM solves a frame of the motion, the markers attached to the model are converted to the ground frame and added to the output motion. This feature is useful if you want to keep a record of the closeness of the fit between the model markers and the recorded markers. It is also useful if your recorded markers are noisy or have missing frames, and you want your output motion to contain complete data for every marker for every frame.

read marker names from

For C3D import only: This option allows you to choose from which parameter field in the C3D file to read the

names of the tracked markers. Because the POINT: LABELS field in a C3D file is limited to four characters, some software packages (e.g., Cortex) store the full marker name in the POINT: DESCRIPTIONS field. Since the marker names in the tracked file must exactly match the names used in the mocap model, if your C3D file does not contain full marker names in the POINT: DESCRIPTIONS field, you may have to edit the mocap model so that the marker names match the four-character names stored in the POINT: LABELS field.

X axis

X axis units, start at zero

These options give you control over the specification of the X axis of the motion that is created from the marker data. The units along the X axis can be either time (in seconds), or frame number. The starting X value of the motion will be 0.0 if the units are time, and 1 if the units are frame number, unless frames of data are cropped because of missing markers (see the *crop ends* option above). For example, if 12 frames of data are cropped from the beginning of the motion (and the data frequency is 60 Hz), the starting X value will be 0.2 seconds for units of time, and 13 for units of frame number. If you want the X values to start at 0.0 (or 1 for frame number) even if frames are cropped, turn on the *start at zero* option.

Analog Data

auto-load analog files

When this box is checked, SIMM will look for and automatically load any analog (ANB, ANC) or XLS data files with the same base name as the TRB/TRC file. If SIMM did not detect the presence of any analog files when the TRB/TRC file was selected, this option is grayed out.

If you selected a C3D file with the file browser, then this box controls whether or not the analog data will be read from the C3D file.

calibrate forces

If an analog file is present, and the *auto-load analog data* box is checked (see above), then this box is active and gives you control over the calibration of the forceplate data. When this box is checked, SIMM determines the baseline of each forceplate channel and automatically subtracts these baseline values from the data, thus "zero-ing out" the force data.

Note: SIMM calibrates forces by finding the frame with the smallest vertical force and assuming that this force should be zero. The analog channel values for this frame are then subtracted from every frame in the motion. If you load a motion during which one or more forceplates has weight on it for the entire motion, then there is no frame which should have zero force, and SIMM will not calibrate correctly. To load such a motion, turn on this checkbox so that SIMM will not perform any calibration of the force channels.

adaptive MVCs for EMGs

When this option is on, SIMM will calculate the maximum value of each EMG channel for the motion trial. If the actual maximum is greater than the MVC specified in *personal.dat* (if any), SIMM will use the maximum as the MVC, instead of the specified value.

use Bayes filter for EMGs

When this option is on, SIMM will apply a Bayes filter to each EMG channel, and then calculate onset time, offset time, and average amplitude from the filtered signal. The calculated values are stored in the motion file in an auxiliary data column that is linked to the EMG column (*e.g.*, "semimem_r" and "semimem_r_aux"). When this option is off, the moving-window smoothing algorithm described below is used instead.

EMG smoothing passes EMG moving window size

These parameters are used to smooth EMG data when the use Bayes filter for EMGs option is turned off. When SIMM uses this type of processing, it rectifies the signal, then smooths it using a moving-average algorithm. The window size for the moving average is specified by the EMG moving

window size field, and the number of times that the algorithm is applied to the data is specified by the *EMG smoothing passes* field.

Save Options

save HTR file

This option allows you to save an HTR file containing the motion that SIMM calculates from the marker data. This HTR file cannot be read back into SIMM, but is useful if you want to import the motion into another software package. If this box is checked, a *browse* button is enabled that allows you to specify the name and location of the HTR file.

save motion file

This option allows you to save a SIMM motion file containing the motion that SIMM calculates from the marker data. This file contains exactly the same data that is in the motion that SIMM loads onto the model. You can load this motion file into SIMM at a later time, rather than re-importing the marker file. If this box is checked, a *browse* button is enabled that allows you to specify the name and location of the motion file.

5.2.3 Analog Data

Analog data files contain forceplate and EMG data that was collected in sync with motion data. When loading C3D files, there are no separate analog files; all of the analog data is contained in the C3D file. When loading TRB/TRC files, you can load analog files only if they correspond to the chosen TRB/TRC files. If the analog file has the same base name as the TRB/TRC file, then the Motion Module will load it automatically when you select the marker file. Otherwise, you should select the analog file as well in the file browser. The same holds for XLS files, which are not actually analog files, but are treated similarly. XLS files can contain other data corresponding to the recorded motion,

such as kinetic data calculated by OrthoTrak and stored in a "single trial spreadsheet."

SIMM can recognize three types of analog data: ground reaction forces, EMG activation levels, and "other" data (usually kinetic data from an XLS file). These data types, and how they are interpreted by SIMM, are described below:

forceplate data

SIMM displays forceplate data by drawing a vector in the model window at the appropriate point of application and with a size corresponding to the magnitude of the force. Forceplate data in an analog file are voltages measured by forceplate transducers. These voltages are converted into forces using a calibration file, *forcepla.cal*. This file is the same one used by Cortex and OrthoTrak. To use it with SIMM, you should put a copy of it in the same folder as your motion data, or in the folder *SIMM\Resources\mocap\misc*. If you have only one forceplate configuration for your motion capture system, it is preferable to put *force-pla.cal* in *SIMM\Resources\mocap\misc*, rather than copying it into every folder of motion data.

Note: C3D files that contain force plate data also contain the calibration information for the plates. Thus there is no seprate calibration file that SIMM reads when importing C3D files.

SIMM also uses another configuration file, *importVariables.txt*, to map forceplate channels to SIMM variables. This file is located in *SIMM\Resources\mocap\misc* (it can also be placed in the folder with your motion data), and contains mappings for typical channel names for up to six forceplates. You will only need to change this file if you use more than six forceplates, or use forceplates that have exotic channel configurations. This file is used when loading ANB/ANC files and when loading C3D files. See Section

5.4.1, importVariables.txt, for information on how to modify this file.

EMG data

SIMM displays EMG data by varying the sizes and colors of the corresponding muscles in the SIMM model. EMG data in an analog file are voltages measured by the EMG system. SIMM rectifies and smooths these data, and then scales them based on an MVC value (maximum voluntary contraction), resulting in a smooth muscle excitation level that varies between 0.0 and 1.0. If MVC values are located in the configuration file importVariables.txt, SIMM will use them to scale the EMG data. If MVC values are not present, SIMM will use each muscle's maximum voltage in the analog file to scale that muscle's EMG data (thus each muscle's excitation will peak at 1.0 sometime during the motion). The file importVariables.txt, located SIMM\Resources\mocap\misc or in the folder with your motion data, contains mappings between typical EMG channel names and the muscle names in the mocap model. It does not contain any MVC values. See Section 5.4.1, importVariables.txt, for information on how to add MVC values to this file. In most cases, however, it is sufficient to not specify them and use SIMM's default scaling method.

other data

"Other" data is contained in XLS files, and can represent any motion variable that you choose to calculate and store in the file. It is usually reserved for kinetic data (e.g., joint moments and powers) that OrthoTrak calculates and stores in its spreadsheet (XLS) format. It may also include motion events, such as toe-off and heelstrike, that are stored at the top of the XLS file. SIMM does not perform any calculations on these data, but does import them so that you can create plots of them in the Plot Maker. SIMM will only import "other" data that are identified as such in import-Variables.txt. This configuration file, located in SIMM\Resources\mocap\misc, contains mappings between

OrthoTrak and the mocap model of all forces, moments, and powers for the hip, knee, and ankle joints. You will only need to edit this file if you want to import data other than these.

5.2.4 Real-time Import

In addition to importing tracked marker files, SIMM can import motion data that is sent over the network in real-time from Cortex. SIMM is thus able to animate a musculoskeletal model and plots of joint angles and muscle lengths while the subject's motion is being recorded. For this real-time connection, Cortex solves tracked marker data using the mocap model. It then sends generalized coordinate values (as well as analog data) over the network to the SIMM computer. If the same mocap model is loaded into SIMM, these generalized coordinates will drive the animation of the model in real-time, with a small delay (whose length depends on the network speed and the graphics speed of the SIMM computer).

Follow these steps to use the real-time connection between Cortex and SIMM:

First-time setup only:

- Find the folder SIMM\Cortex on your SIMM computer and look for GaitModel.jnt and solver.dll (solver.dll may be hidden in the folder view because it is a system file). Copy both files to the folder on the machine where Cortex resides (you'll need to exit Cortex first, if it is running). This will guarantee that Cortex is using the same mocap model and the same scaling algorithms as SIMM uses.
- Open the text file SIMM\Resources\preferences.txt in a text editor such as Notepad or Wordpad. Locate the line that reads: CORTEX_MACHINE <hostname>, and change the host-

name to the name (or IP address) of your Cortex machine. Save and close the file. You can also set the CORTEX_MACHINE preference in the Scripting Tool, and then save the SIMM preferences so they will be restored the next time you launch SIMM (see Section 2.18 for details).

If your motion capture system includes forceplates, copy the file *forcepla.cal* from your Cortex computer onto your SIMM computer and put it in the folder *SIMM*\Resources\mocap\misc.

Each motion capture session:

- © Copy the folder containing the motion data from the Cortex computer to the SIMM computer (or make it shared). If there is a *personal.dat* file for this data, make sure it is in the folder too.
- Launch Cortex and load the appropriate capture file. This will load the setup (.cal), tracks (.trb/.trc), and analog (.anb) data for the capture.
- Select File -> Skeleton Definitions -> Create SIMM Calcium Skeleton (JNT File). Make sure the JNT file is set appropriately (the default is mocap.jnt) and the static pose is set to the proper static tracks file. Then click Create.
- Launch SIMM.
- Open the Mocap Model by dragging the static trial marker file from a Windows file explorer and dropping it onto the SIMM background window. You can also open it using *File... Open Mocap Model* in the menu bar and browsing for the marker file.
- Set the options as desired in the dialog box and click OK.
- Open the Model Viewer window.

- In the Model Viewer window, choose *start* > *realtime connection to* <*hostname*>. SIMM will display a dialog box allowing you to set some options for the connection. The *motion buffer size* options control how many seconds of motion data are saved in SIMM's buffer. The *time scale* options let you specify the minimum and maximum values, in seconds, for the time scale of the motion. If you want the scale to remain fixed between *minimum* and *maximum*, check the *sliding* checkbox, otherwise the scale will continue to increase as new data is received.
- SIMM will now wait to receive data from the Cortex computer. Once the connection is established, SIMM will display "connected" in its message window, and the SIMM model will begin tracking the motion of the subject in real-time. You can pan, zoom, rotate, and change the draw mode of the SIMM model as it is tracking the motion. You can also create plots of kinematic variables and muscle properties and see the plots change in real-time.
- To disconnect SIMM from the real-time stream, click the *stop* button in the Model Viewer. You can play back the last N seconds of the motion.

Note: When analog data is imported into SIMM in realtime, it is processed slightly differently than when the data is post-processed in SIMM. This is because the real-time analog data is processed frame-by-frame, without the benefit of the full data set. This has the following implications:

- * To set the baseline for the forceplates, the first frame of force data is used as the "zero" level. Thus when you first connect SIMM to Cortex, you should make sure that nothing is on the forceplates.
- * If no MVC levels have been specified for some of the muscles, a running tally of each muscle's maximum

level will be kept, and used to scale the EMG signals into the range 0.0 to 1.0. Thus if you want to accurately scale EMG levels throughout a real-time SIMM motion import, you should either specify MVC levels, or have the subject perform MVCs just after connecting SIMM to Cortex.

5.2.5 IKS Files

IKS files are a special type of input file for the Motion Module. They are identical in format to motion files (MOT), but they can be used as input to the Motion Module, similar to TRC, TRB, and C3D files. The unique feature of IKS files is that you can use them to specify gencoord values as inverse kinematics targets in addition to marker locations. That is, for each frame of input data, the IK algorithm will optimize the gencoords of a model to best fit the marker coordinates and the gencoord values specified in the IKS file.

If the IKS file contains a column of gencoord values and a corresponding column of gencoord weights using the auxiliary data feature as described for motion files in Section 3.5.6 (e.g., "knee flexion r" and "knee flexion r aux"), the specified gencoord values will be used as targets during inverse kinematics. The least-squares algorithm will attempt to keep the gencoord close to its specified value, using a penalty determined by the gencoord weight. If the weight of a gencoord for a given time frame is greater than 1000.0, the IK algorithm will lock the gencoord at the specified value instead of solving for it. Because the auxiliary column contains a weight value for each time frame, you can vary the gencoord weight from frame to frame. You can also use this auxiliary mechanism to specify frame-specific weights for the marker coordinates in the IKS file. If you do not specify weights for a column of marker coordinates, the marker's

weight in the model will be used. However, if you do not specify weights for a column of gencoord values, the gencoord values will be ignored during IK.

Note: IKS files do not support the loading of analog data from ANB or ANC files. To include EMG or force-plate data in a motion created from an IKS file, the processed analog data must be in the IKS file itself. A typical use case for IKS files would be the following. Start with a TRC/ TRB/C3D file and import it onto a model, saving a motion file (MOT) during the process. During the import, turn on the show markers box to include the marker coordinates in the MOT file, and make sure the analog data is loaded and processed as well. The resulting MOT file will contain [IKsolved] gencoord values, as well as marker coordinates, muscle EMG, and force-plate forces. Modify the file in Excel, focusing on the columns of gencoord data. Delete any gencoords that you do not want to specify as input. For the others, modify their values as desired, and add auxiliary columns with their weights. Save your changes and give the file the IKS suffix. You can now import it into SIMM and perform IK using the specified gencoords and markers.

5.3 Using the Mocap Model

As described in Section 5.2, SIMM has the ability to read tracked marker data and convert it into a motion by fitting a musculoskeletal model to it. For this to work well, the body segment lengths, marker names, and marker locations in the model must exactly match those for the subject whose motion is being recorded. Because it is time consuming to measure and scale the body segments, and measure and record the offsets of all of the markers, the Motion Module has the ability to automatically scale a pre-made model (the *mocap model*) to fit the subject.

To use the mocap model, drag and drop the static pose marker file onto the SIMM background window or use File... Open Mocap Model in the menu bar and browse for the marker file. This static pose (described in more detail in Section 5.3.2) is used to calculate joint center locations and segment lengths for the subject, using the same algorithms implemented in OrthoTrak. In other words, the Motion Module recreates the OrthoTrak skeletal model from the static pose, and then maps this skeletal model onto the mocap model. Thus to use the mocap model, you need to use the same motion capture protocol as you would for OrthoTrak. You can use either the Helen Hayes or Cleveland Clinic marker sets (plus your own additional markers, if desired), as long as the marker names and locations match the protocol defined in the OrthoTrak manual. The Motion Module uses the tracked marker data from the OrthoTrak static pose, and also segment information from personal.dat, to scale the mocap model to the subject. The algorithms for calculating joint center locations and segment lengths have been designed to be as similar as possible to the OrthoTrak algorithms. This was done so that motion information in SIMM (e.g., joint angles, EMG levels) would match the corresponding information in OrthoTrak, and also so that you would not have to change your OrthoTrak protocol in order to use SIMM. For more information on the markers built into the mocap model, and how to add your own, read the Guide to Mocap Model Markers document.

The mocap model and the algorithms used to scale it are described in the following sections.

5.3.1 The Mocap Model

SIMM comes with several musculoskeletal models that can serve as the *mocap model* in the Motion Module. The

default model is a full-body model that has been customized for gait analysis, but can be used to import and display any type of full-body motion. The model has 41 body segments, 41 joints, 40 degrees of freedom, and 88 lower-extremity muscles. It represents an adult male, approximately 175 cm. tall, with a mass of 78 kg. The model is scaled to match the size of the motion capture subject using algorithms described in Section 5.3.4. The model's joints have been carefully constructed to represent normal joint motion as closely as possible. This model also includes mass and interia properties for all of the body segments, so it can be used by the FIT Module or Dynamics Pipeline for inverse and forward dynamic simulations. The joint file for this model is SIMM\Models\GaitModel\GaitModel.jnt. There are other [legacy] mocap models located in SIMM\Resources\mocap.

SIMM keeps track of which *mocap model* to use by storing a pointer to the appropriate joint file. When SIMM is first launched, it initializes this pointer by reading the MOCAP_MODEL variable in the preferences file (SIMM\Resources\preferences.txt). The default setting for this variable is SIMM\Models\GaitModel\GaitModel.jnt. If you want to change the *mocap model* while SIMM is running, click on *Options* in the menu bar and select *Choose Mocap Model*. You can then browse for the joint file of the desired model.

You may change any of the *mocap models* however you wish. For example, you can add or remove muscles from the model, or change the tendon and fiber parameters of existing muscles. You can also add degrees of freedom to the model, in order to more accurately represent a particular motion (*e.g.*, adding toe joints and gencoords to examine toe motion in greater detail). If you modify the *mocap model*, however, you should keep in mind two things.

First, the model has been set up to correspond to the skeletal model that OrthoTrak uses when processing gait data. The lower-extremity body segments and orientations of the reference frames closely match those in the OrthoTrak model. Also, each body segment in the mocap model is scaled to fit the subject by relating its length to the length of an OrthoTrak segment. These relations are specified in the mocap model by defining scale segments and scale factors for each body segment. If you add, delete, or modify joints or body segments in the mocap model, you should make sure that each segment still properly relates to an OrthoTrak segment.

Second, *GaitModel.jnt* contains several macros that are used to properly define the orientation of the floor, and to automatically remove the upper body segments if there are no upper body markers. When SIMM reads a joint file, it performs these macros but does not save them internally. Thus when it writes out a joint file, all of the macros have been removed. If you make changes to the mocap model in SIMM and then save the new model to a file, do not replace *GaitModel.jnt* with the new file. Instead, copy the relevant portions of the new file into *GaitModel.jnt* using a text editor, thus preserving the macros and comments.

5.3.2 The Static Pose

When you open the mocap model, SIMM prompts you for the name of a tracked marker file containing a static pose of the subject. This static pose is the same one used by OrthoTrak, and for it you can use any of the six marker sets identified by that software package: Cleveland Clinic Lower Body, Cleveland Clinic Full Body, Cleveland Clinic Full Body with Head, Helen Hayes Lower Body, Helen Hayes Full Body, and Helen Hayes Full Body with Head. It is also strongly recommended that you include the medial knee and

ankle markers in the static pose, for more accurate calculation of knee and ankle joint centers. You can also supplement the OrthoTrak marker set with your own custom markers, as long as you do not move or remove any markers from the identified set. Lastly, the marker set used in the static trial must include all of the markers you plan to use for capturing motion. This is because the Motion Module calculates the locations of all markers in the mocap model based on their locations in the static trial. These are the steps you should follow when collecting the static trial:

- Choose which of the six OrthoTrak marker sets you would like to use for capturing motion.
- Add the medial knee and ankle markers, for better calculation of knee and ankle centers (not required, but highly recommended).
- Add any additional markers that you would like to use (e.g., extra markers on the feet, more markers on the arms). These markers must also be added to the mocap model, as described in Section 5.3.5, The Marker Set.
- Capture the static trial using the protocol outlined in the OrthoTrak manual. The subject should have their arms either down by their sides, or straight out from their body with their thumbs facing forward.
- Remove the medial knee and ankle markers, and any others that you do not want to use for capturing motion.

Note: If you use a marker set with no upper extremity markers, the Motion Module will remove the upper extremity from the mocap model and display only the pelvis and legs.

Once you have selected the static pose file to be used for opening the mocap model, SIMM displays a dialog box, allowing you to set some options for importing the static

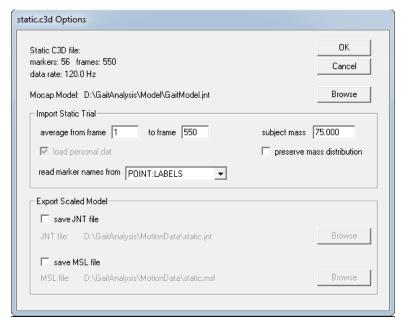


Figure 5-2. Static Trial Import Dialog Box

pose. In many cases, you will want to use the default settings for these options, so you can simply click the OK button to import the motion. The following list describes each option in the dialog box:

average from frame, to frame

These fields allow you to specify the starting and ending numbers for the sequence of frames that are averaged together to determine the static pose. These fields are initialized to 1 and the number of frames in the file, meaning that all frames will be averaged. If frames in the chosen sequence are missing some markers, locations for markers that are present will still be used in the average.

subject mass

This field allows you to specify the mass of the motion capture subject. The units should be the same as the units used in the mocap model. If the subject mass is specified in *personal.dat* (in the field *Weight:*), that value will be used to initialize the field in the dialog box.

preserve mass distribution

This field specifies whether or not the mass distribution in the [unscaled] mocap model should be preserved in the scaled model (the alternative is to scale the mass of each body segment based on its geometric scaling). For more information on scaling mass parameters, see Section 5.3.4, Scaling the Mocap Model. Whichever method you choose, the final mass of the scaled model will always be equal to the mass specified in the *subject mass* field; only the distribution of that mass among the various body segments will be different.

load personal.dat

This option gives you control over the automatic loading of *personal.dat*. When SIMM loads the static marker file, it looks for a file called *personal.dat* in the same folder. This file is identical to the one created and used by OrthoTrak. If the file is present, SIMM will automatically load it and read model parameters from it, such as foot length and hip origin offsets. It will use these parameters to determine joint center locations and segment lengths, using the same algorithms that OrthoTrak does. If there is no *personal.dat* file present in the folder, this option will be grayed out. If it is checked and you do not want to load *personal.dat*, click the box to turn it off.

read marker names from

For C3D import only: This option allows you to choose from which parameter field in the C3D file to read the names of the tracked markers. Because the POINT:LABELS field in a C3D file is limited to four characters, some software packages (e.g., Cortex) store the full marker name in the POINT:DESCRIPTIONS field. Since the marker names in the tracked file must exactly match the names used in the mocap model, if your C3D file does not contain full marker names in the POINT:DESCRIPTIONS field, you may have to edit the mocap model so that the marker names match the four-character names stored in the POINT:LABELS field.

save JNT file, save MSL file

These options allows you to specify if SIMM will write out joint and muscle files containing the musculoskeletal model that is scaled to fit the subject. After SIMM has loaded the mocap model and scaled it based on the data in the static marker file and *personal.dat*, it will write out corresponding joint and muscle files, depending on the states of these check boxes. You may want to create these files so that you can make changes to them or to be able to re-load the model without going through the scaling process again.

5.3.3 Calculation of Joint Centers

Once the static pose has been loaded, the Motion Module recreates the OrthoTrak skeletal model from the marker cloud. The first step is determining the locations of the joint centers for all of the joints in the OrthoTrak model. The pelvis, hip, knee, and ankle centers are all found using the same procedure used by OrthoTrak.

The hip center is determined using percentage offsets from the pelvis markers. The values are percentages of the distance between the left and right ASIS markers. The Motion Module reads these offsets from *personal.dat*, as written by OrthoTrak. The default values for these offsets are taken from *Bell et al. Journal of Biomechanics*, 23(6), 1990, pp. 617-21:

```
Right Hip Posterior Displacement (%): 22.00
Right Hip Lateral Displacement (%): 32.00
Right Hip Inferior Displacement (%): 34.00
Left Hip Posterior Displacement (%): 22.00
Left Hip Lateral Displacement (%): 32.00
Left Hip Inferior Displacement (%): 34.00
```

To change these values, edit the file *personal.dat*, as described in Appendix D of the OrthoTrak manual.

The knee and ankle centers are found using the medial and lateral markers. It is strongly recommended that you use medial markers for a more accurate calculation of joint centers. If you choose not to use them, you should enter knee and ankle diameter measurements into *personal.dat*. The Motion Module will use them to locate the knee and ankle centers if no medial markers are used.

The default method for determining shoulder, elbow, and wrist joint centers uses percentage offsets from the appropriate marker locations. If medial elbow and wrist markers are used in the static trial, their locations are averaged to get the joint centers, as is done with the knee and ankle. It is recommended that you use medial elbow and wrist markers in the static trial if you want an accurate representation of arm motion. Calculation of shoulder joint centers uses the same technique as for hip center. The reference length is the distance between the shoulder and elbow markers. The offsets are specified as percentages of this distance. The default values are shown in the example below:

```
Right Shoulder Inferior Displacement (%):12.5
Right Shoulder Anterior Displacement (%):-1.25
Right Shoulder Lateral Displacement (%):-2.5
Left Shoulder Inferior Displacement (%):12.5
Left Shoulder Anterior Displacement (%):-1.25
Left Shoulder Lateral Displacement (%):-2.5
```

5.3.4 Scaling the Mocap Model

Once the locations of the OrthoTrak joint centers have been calculated from the static pose, the Motion Module determines the orientations of the OrthoTrak segment reference frames. It then can measure the lengths of the OrthoTrak segments and use them to scale the mocap model to match the size of the subject.

The reference frames for the foot, shank, thigh, pelvis, and torso are all determined using the procedure described in Appendix H of the OrthoTrak manual.

OrthoTrak does not create reference frames for the upper and lower arms, but the Motion Module does this using one of several methods. If medial elbow and wrist markers are used in the static trial, then the arm reference frames are found in the same way in which the thigh frames are found. If no medial markers are present, then the upper and lower arm reference frames are found using the line between the joint centers as the X axis, and using the same Y axis as the torso. The Z axis is then determined by crossing X and Y.

Once all of the segment reference frames have been determined, the length of each segment is calculated. For most segments, the length is simply the distance from one joint center to the next. For the foot, the Motion Module reads the length from *personal.dat*. If there are no foot length measurements in *personal.dat*, then the foot length is assumed to be 1.4 times the distance from the heel marker to the toe marker (the toe marker is actually placed on the top of the foot just posterior to the toes).

Each body segment in the mocap model contains scaling information that tells the Motion Module how to scale it based on an OrthoTrak skeletal segment. The scaling information consists of the name of the OrthoTrak segment and X, Y, and Z reference numbers that correspond to the unscaled length of the SIMM segment. For example, the right femur in the mocap model contains the line:

```
gait scale R THIGH 0.3960 0.3960 0.3960
```

This tells the Motion Module that the unscaled femur is 0.3960 meters long. Once the length of the corresponding OrthoTrak segment is known (R_THIGH), the femur can be scaled accordingly. If the R_THIGH segment were 0.35 meters long, then the femur would be scaled by a factor of 0.35/0.396. In most cases the three reference values are the same number, indicating that the segment should be scaled

uniformly in X, Y, and Z. The two exceptions are the TORSO and PELVIS, which are scaled differently in two dimensions. For SIMM segments that do not map directly to an OrthoTrak segment, their scaling information is copied from the most relevant segment. For example, the right hand in the mocap model copies the scaling information from the right lower arm, so that the hand is scaled the same amount as the lower arm.

5.3.5 The Marker Set

The marker set in the full-body mocap model that comes with SIMM includes every marker used in all six marker sets that OrthoTrak recognizes, plus the medial knee and ankle markers. In addition, many other markers have been added, such as medial elbow and wrist markers. For a complete list of the markers in the model, as well as information on when they should be used and where they should be placed on the subject, read the Guide to Mocap Model Markers document (MotionModuleMarkers.pdf, located in SIMM\Documentation). The mocap model contains over 80 markers, which is more than the number used in most applications. When the static trial is loaded, any marker in the mocap model which is not in the static trial is removed from the model. Thus it is not a problem to have extra markers in the mocap model. In fact, you should add to the model whatever extra markers you may need for any of your motion capture applications. Then for a particular application the mocap model will have all the necessary markers, and the unused ones will automatically be removed when the model is loaded into SIMM.

To add or change markers in the mocap model, use the Marker Editor, which is described in Section 2.12. As discussed in Section 5.3.1, you should be careful not to overwrite the original *GaitModel.jnt* file. Instead, after editing

the marker set, save the model to a new file name, and copy the altered markers into *GaitModel.jnt*.

All of the markers in the mocap model have X, Y, and Z offsets that put them in realistic locations given the dimensions of the generic model. Thus if you load the unscaled mocap model into SIMM (by dragging it from a Windows file explorer and dropping it onto SIMM, or using the *File... Open* command in the menu bar), the markers will appear in positions corresponding to where they are placed on the subject. *These offsets are purely decorative, to help you view the marker set. They are not used by the Motion Module to process any marker data.* To explain why this is so, we must first introduce the concept of *critical* markers and *non-critical* markers.

Critical markers are ones that must be present in the static trial in order for the Motion Module to load and scale the mocap model. For the lower body, these markers are: V.Sacral, R.ASIS, L.ASIS, R.Knee (or R.Knee.Lateral), R.Ankle (or R.Ankle.Lateral), R.Heel, R.Toe, L.Knee (or L.Knee.Lateral), L.Ankle (or L.Ankle.Lateral), L.Heel, and L.Toe. If any of these markers is missing from the static trial, the SIMM model of the lower body will not be loaded. For the upper body, the critical markers are: V.Sacral, R.ASIS, L.ASIS, R.Shoulder, R.Elbow, R.Wrist, L.Shoulder, L.Elbow, L.Wrist. Non-critical markers are all other markers in the set. Note: starting with SIMM 4.0, the critical marker set has been modified slightly. For example, instead of using the sacral marker, V.Sacral, you may use the right and left PSIS markers, R.PSIS, and L.PSIS. Consult the Guide to Mocap Model Markers for complete details on which combinations of marker sets are required for all of the mocap models.

Once the Motion Module has determined the locations of the joint centers and the orientations of the segment reference frames from the static pose, it calculates the proper offsets for all of the critical markers (plus the static-only medial markers). For example, once the right thigh reference frame has been oriented within the static pose marker cloud, the exact positions of the critical markers attached to the right thigh can be measured directly from the static pose and entered into the mocap model, thus overwriting whatever offsets were in the model input file.

After the offsets of all the critical markers have been determined in this fashion, the mocap model is "fit" to the static pose marker cloud (as described in Section 5.1) using only the critical markers to find the best fit. This process orients the mocap model within the marker cloud, so that the offsets of the non-critical markers can be measured directly from the static pose. These offsets are then entered into the model, overwriting whatever values were in the model input file

To summarize, the Motion Module uses a two-step process to calculate proper offsets for all of the markers in the mocap model. The first step determines the offsets of the critical markers, which the OrthoTrak algorithms can definitively locate without knowing anything about the mocap model. Then these critical markers are placed on the mocap model, and the model is fit to the static pose marker cloud. Now the offsets of the other markers can be measured, because every body segment in the mocap model is now correctly placed in the static pose.

5.4 Analog Configuration Files

SIMM can include analog data such as ground reaction forces, EMG activation levels, and kinetic data when importing a motion. SIMM uses a configuration file named importVariables.txt to determine which analog variables to import from an analog file, and how the data for each variable should be interpreted. This configuration file is used for both TRB/TRC import (with corresponding ANB/ANC analog files) and for C3D import (where the analog data is contained in the C3D file itself). SIMM can interpret analog data as one of three types:

forceplate data

These variables specify voltages representing force or moment components as measured by a forceplate transducer. Given the voltages generated by a forceplate (6 channels for an AMTI or Bertec forceplate, 8 channels for a Kistler forceplate) SIMM can calculate and display a force location and vector for the forceplate.

EMG data

These variables define activation levels for one or more muscles in the SIMM full-body model. SIMM rectifies, smooths, and scales EMG data so that it can be plotted, and used to control the width and color of muscles during an animation.

other data

Any variables that are not forceplate or EMG data are classified as other data. SIMM does not perform any calculations on these data variables, but they may be included in SIMM plots.

5.4.1 importVariables.txt

The file *importVariables.txt* file contains a list of variable names and attributes. When SIMM processes an analog data file or an OrthoTrak XLS file, it consults the *importVariables.txt* file to decide which variables to import and how to interpret them. The file can also contain alternate definitions of the critical markers in the mocap model. These are the markers that have to be included in your motion capture

marker set in order for SIMM to automatically scale the model to fit the subject.

The default version of this file is located in *SIMM\Resources\mocap\misc*. You can customize the file and leave it in this folder, or you can put a copy in the folder containing your motion data. When SIMM processes a motion trial, it first looks in the motion data folder for *importVariables.txt*. If it is not found there, SIMM then looks in *SIMM\Resources\mocap\misc*.

Each row in *importVariables.txt* defines a variable to be imported. The first column in a row specifies the name of the variable as it appears in the analog or XLS data file. Since certain analog files support variable names with spaces in them, the first column of the *importVariables.txt* file *must* be terminated by a tab character. SIMM considers all characters from the beginning of a row until the first tab character to be the name of the import variable. SIMM does a case-insensitive comparison when matching variable names defined in *importVariables.txt* with variable names in an analog data file. Therefore the name "Rt Tibialis" would be considered the same as "rt tibialis".

The second column in a variable definition specifies the type of the variable. It must be one of the following keywords: force_plate, muscles, marker, or other_data. These keywords must be lowercase. Following each keyword is information describing the variable:

force plate

This keyword specifies a ground reaction force variable. It must be followed by three values:

- 1. The forceplate number (1, 2, 3, etc.), then
- 2. The keywords force or moment, then

3.

The channel component (x, y, or z for AMTI or Bertec forceplates, or x12, x34, y14, y23, z1, z2, z3, z4 for Kistler forceplates).

For example, a 6-channel AMTI or Bertec forceplate mapping for forceplate 1 would look like this:

```
F1X force_plate 1 force x
F1Y force_plate 1 force y
F1Z force_plate 1 force z
M1X force_plate 1 moment x
M1Y force_plate 1 moment y
M1Z force plate 1 moment z
```

An 8-channel Kistler forceplate mapping for forceplate 2 might look like this:

```
F1X1+2
         force plate 2 force x12
F1X3+4
         force plate 2 force x34
F1Y1+4
         force plate 2 force y14
         force plate 2 force y23
F1Y2+3
F1Z1
         force plate 2 force z1
         force plate 2 force z2
F1Z2
         force plate 2 force z3
F1Z3
F1Z4
         force plate 2 force z4
```

muscles

This keyword specifies an EMG variable. It must be followed by one or more SIMM muscle names. The keyword mvc may optionally appear after the last muscle name. If mvc appears, then it must be followed by an integer number that SIMM uses as the voltage (usually in millivolts) for the maximum voluntary contraction when scaling that EMG channel. If no MVC value is specified, then the channel is scaled such that its maximum value is 1.0. EMG scaling is performed *after* the EMG channel's data has been smoothed and resampled to the motion's frequency. Here are some example channel mappings:

```
Rt rectus muscles rectus_fem_r mvc 500
Rt tibialis muscles tib_ant_r mvc 735
Rt gastroc muscles gas_lat_r gas_med_r
```

Rt hamstring muscles semimem_r semiten_r

marker

This keyword specifies a critical marker. SIMM uses critical markers to measure the lengths of the body segments in the static trial, in order to scale the mocap model to fit the subject. The critical markers must be given special names so that SIMM can recognize them, and SIMM has several built-in names for each marker. For example, the built-in names for the right ASIS marker are R.ASIS, RASIS, RASI, and RILI. If you use any one of these names for the right ASIS marker in your motion data, then SIMM will automatically recognize it as the appropriate critical marker. If you do not want to use one of those names, or you have numerous motion trials that have already been captured and processed using a different marker name, you can define a mapping between your name and SIMM. The mapping consists of the name of your marker, followed by the keyword marker, followed by the SIMM keyword for that marker. For example, to use *right.asis* as the name of the right ASIS marker, use this mapping:

right.asis marker RASIS

For a complete list of the built-in names for each critical marker, see the document *MotionModuleMarkers.pdf*, located in the folder *SIMM\Documentation*. For a complete list of the SIMM keywords for each critical marker, look at the bottom of *importVariables.txt* in *SIMM\Resources\mocap\misc*.

other_data

This keyword specifies a data channel that exists simply to be included in SIMM plots. This keyword may be optionally followed by a single word that will be used to label this channel in SIMM plots. If no name follows the other_data keyword, then the name of the imported variable will be used.

5.4.2 forcepla.cal

When importing analog data from ANB/ANC files, SIMM uses the same calibration file as Cortex and OrthoTrak for processing forceplate data. Therefore, you can simply copy the *forcepla.cal* file from your Cortex folder into the *Resources\mocap\misc* folder. For users who need to create a *forcepla.cal* file to describe their forceplate(s), refer to Appendix C of the OrthoTrak manual.

Note: *forcepla.cal* is not used for C3D import since C3D files contain the necessary calibration information for the force plates.

Appendix A: Tips on Using SIMM

The following is a list of tips on using SIMM. It is strongly recommended that you read through the entire list before using SIMM for any of your work.

- Drag and drop is a very handy method of loading files into SIMM. Every type of file that SIMM can read can be dropped onto the SIMM window to load it. Read Section 2.2.1, Drag and Drop, for a description of the rules that SIMM follows when a file is dropped onto one of its windows. Drag and drop is particularly useful for loading models (simply drop a JNT or SMM file onto SIMM), and loading motion capture files (drop a TRB/TRC/C3D file onto the SIMM background to load it as a static trial, and onto a model window to load it as a motion trial).
- Use the SIMM hot keys. Ten of the function keys work as hot keys in SIMM, as described in Section 2.4.6, Model Window Viewing Commands. They can be very handy for turning on or off muscle and marker displays, or for stepping through a motion frame by frame. Annotation mode (F11) can be particularly useful when you want to identify an object in the model display.

- Saving joint and muscle files: Many joint and muscle files, including all of the models that come with SIMM, contain comments and macros that are not preserved when the model is loaded into SIMM and then saved back to a file. For this reason, you should be careful when saving a joint or muscle file not to overwrite an existing file. Instead, save the data to a new file, and then use a text editor to copy the modified model elements into the original joint or muscle file.
- Use model archives. A model archive is a compressed file containing all of the files associated with a model, including joint file, muscle file, bones files, and any motion files currently linked to the model. It contains everything needed to load the model back into SIMM at a later time. Because everything is in one file, it is a good way to share models with other users, or to backup your models before modifying bone files or other components that are typically shared among models.
- Live plot curves are very helpful when creating or modifying models (see Section 2.9.2, Live Plot Curves, for details on how to create them). For example, if you make a plot curve of a muscle moment arm as a function of a generalized coordinate, and make the curve live, it will be automatically updated as you move muscle attachment points, move wrap objects, modify joint parameters, or make any other change that affects the muscle's moment arm. Live plot curves are particularly helpful when positioning wrap objects. because they enable you to see the wrap object's affect over the entire range of motion, not just in the model's current pose. When using live curves, it is often helpful to create the same muscle curve twice

- and make one of them live, so you can easily compare the current state of the model to the original state.
- The **lookat** command in the Model Viewer is a good way to focus the model display on a particular body segment or muscle. By default, the view is centered on the origin of the ground segment, so when you rotate the scene it will rotate about this point. This default setting can make it difficult to closely examine other areas of the model. For example, to look closely at the toes, you might want to zoom in on that body segment, then right-click on a toe bone and select "look at toe segment" from the pop-up menu. Then when you rotate the view it will rotate about the origin of the toe segment.
- You can reload the preferences file at any time by choosing Options > Reload Preferences File from the menu bar. You can also change preferences using the Scripting Tool and then save them back to the file by choosing Options > Save Preferences File.
- Motion Module: If you load motion trials and the model does not seem to track the markers accurately or be scaled properly, you might want to examine the static trial more closely. After the mocap model has been scaled and loaded, you can re-load the static trial as if it were a motion trial (e.g., using File > Open... instead of File... Open Mocap Model...). This allows you to check the show markers box, so you can replay the static trial and see how well or poorly the tracked markers match the model markers.
- SIMM has a preference called MULTIPLE SCREENS that allows you to take advantage of multiple screens (monitors) on your computer. If this preference is on,

the SIMM tool windows are created as separate toplevel windows. They can be moved outside of the main SIMM window, and they will have their own icons in the Windows task bar. The model and plot windows are still constrained to the main SIMM window. This option works best if you keep the main SIMM window on one screen and the tool windows on another screen, because when you click anywhere in the main SIMM window, or cause a new model or plot window to open, the main SIMM window is "popped" above all of the tool windows. This preference cannot be changed while SIMM is running; it is checked only when SIMM is first launched.

- You can right-click on bones in the model window to display a pop-up menu with options for changing the bone's drawmode, material, and other display options.
- You can right-click on muscles and on muscle points to display a pop-up menu with display as well as editing options.
- Use muscle groups. If you have more than about 15 muscles in your model, you will find it much easier to select the ones you want if they are divided into several groups. In the lower-extremity model provided with SIMM, the muscles are grouped by function. You could also group them by name, the number of joints they span, or any other criterion that you want.
- Use body segment groups and generalizedcoordinate groups to make large models easier to navigate and display.
- Try to keep muscle names short. When SIMM reads in a muscle file, it scans for the longest muscle

- name, then makes every muscle group menu wide enough to hold that name. This is done so that it is easy to position many group menus at once in a window. To avoid large menus that run beyond the window borders, try to limit muscle names to 10 letters.
- In addition to selecting **delete** from the Plot Maker command menu, there is another way to delete plot curves. Pressing the Delete or the Backspace key in a plot or plot key window will delete all of the selected curves from the plot.
- To delete a model, click the close box in the upper right corner of the model's window, or press the Delete or Backspace keys in the model's window.
- If the display of your model is too slow (e.g., for smooth animation), try changing the drawing mode for some of the body segments. For small polygons, flat shading is often indistinguishable from gouraud shading, but is usually faster. For example, when viewing an entire human skeleton, you can probably use *flat shading* for the hand and foot bones without affecting the quality of the display a great deal.
- When displaying shadows of the body segments, you usually want the shadows to be cast on a world object, such as the floor or a wall. When setting the locations of the shadows and world objects to achieve this effect, you should position the shadow a small distance away from the world object so that you can see it. For example, if you position the floor with a Y-coordinate of -0.9, you should position the shadows at [about] -0.895 along the Y-axis. If you put them at exactly -0.9, they would be coincident with the floor, and you may or may not see them, depending on the viewing angle.

- When a body segment or world object is displayed in wireframe mode, the color that is used to draw the object is taken from the *ambient* color component of the material assigned to that object. For the def bone material, this color is not white, but rather a light gray. If you want pure-white wireframe objects, you should define a new material with an ambient color of 1.0 1.0 1.0, and assign it to the object you want to display in wireframe mode.
- If you press a key or select a menu item and nothing seems to happen, you may have lost the input focus to that window. This means that your key presses are either "going" to another window, or getting lost entirely. This can happen when you press a lot of keys while moving the cursor into several different windows. To fix it, just release all the keys and mouse buttons, click the left mouse button on a different window, and then click it again over the window you were originally working in.
- Whenever SIMM pops up a confirm window, you can use keyboard keys as well as the right mouse button to confirm the action. Pressing y will select the yes box, and pressing n or c will select the cancel box.
- You can copy and paste text from the Windows clipboard into text and number fields in SIMM, as well as into the Scripting window.

Appendix B: SIMM Resources

The SIMM installation procedure creates a directory named *Resources* within the directory where SIMM is installed. For SIMM to execute properly the SIMM Resources directory must not be moved or renamed. SIMM needs to access several files within the Resources directory while it runs. Some of these files can be used to modify the way SIMM works. The following sections describe these files.

B.1 Preferences

The file *preferences.txt* contains a set of parameters that you can define to change the behavior of various features in SIMM. The file is read by SIMM when the program is launched, and it can also be reloaded while SIMM is running by selecting *Options... Reload Preferences File* from the SIMM menu bar. When SIMM reads the preferences file, it first looks for it in your user application data folder (e.g., C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\7.0\pref*erences.txt*). If it is not found there it looks in the resources folder where SIMM was installed (e.g., C:\Program Files (x86)\MusculoGraphics\SIMM70\Resources\preferences.txt). edit the preferences while SIMM is running, select *Options...* Edit Preferences. You can also save the current set of preferences back to the file using Option... Save Preferences File. Individual preferences can also be defined or changed using the Scripting Tool (see Section 2.18 for details). Here is a description of each preference in SIMM:

GENERAL PREFERENCES

SIMM FOLDER

This preference specifies the folder containing the SIMM executable program. It is used to help set the values for other preferences if they are not [fully] specified. For example, if the MOCAP_MODEL preference is set to a relative path name, then it is assumed to be relative to the SIMM_FOLDER. In general you should not specify SIMM_FOLDER in your preferences file, instead letting SIMM define it at start-up to be the folder containing SIMM.exe.

RESOURCES FOLDER

This preference specifies the folder containing the primary SIMM resource files, such as *password.txt* and *preferences.txt*. In general you should not specify this preference in your preferences file, instead letting SIMM define it at start-up to be SIMM FOLDER/Resources.

OUTPUT FILE FOLDER

This preference specifies the destination folder for the Plot Maker when creating a muscle orientation file and for the Bone Editor when saving a bone file. The default value is ".", which means that the current working folder will be used. The current working folder is usually the last folder from which you loaded a file or to which you saved a file, so in general you should not override this default.

CORTEX MACHINE

This preference specifies the name (or IP address) of the machine running Cortex to which you want to connect in real-time.

READ_STARTUP_SCRIPT

When this option is on, SIMM will read a start-up script each time it is launched. The file is named SIMM.ssc and is located in your user application data folder (e.g., C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\7.0\SIMM.ssc), so each user can have his or her own start-up script. The file can contain commands to open and position tool windows, as well as load models, motions, and plots. If you customize this file by entering your own commands, be sure to set

WRITE_STARTUP_SCRIPT to *no* so that your modifications are not overwritten when you exit SIMM.

WRITE STARTUP SCRIPT

When this option is on, SIMM will write a script each time the program is closed. The file is named SIMM.ssc and is written to your user application data folder (e.g., C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\7.0\SIMM.ssc). If the READ_STARTUP_SCRIPT preference is set to yes, this file will be read by SIMM the next time the program is launched. The file contains commands to open, close, and position tool windows so that the set up of the windows is preserved from one use of SIMM to the next.

SET TOOLS TO NEW MODEL

This preference specifies whether or not to set all of the SIMM tools to a given model when that model is first loaded.

DISPLAY PREFERENCES

MULTIPLE_SCREENS

This preference specifies whether or not SIMM utilizes multiple screens (monitors) on your computer. If this preference is on, the SIMM tool windows are created as separate top-level windows. They can be moved outside of the main SIMM window, and they will have their own icons in the Windows task bar. The model and plot windows are still constrained to the main SIMM window. This option works best if you keep the main SIMM window on one screen and the tool windows on another screen, because when you click anywhere in the main SIMM window, or cause a new model or plot window to open, the main SIMM window is "popped" above all of the tool windows. This preference cannot be changed while SIMM is running; it is checked only when SIMM is first launched.

FASTER MUSCLE DRAWING

This option causes SIMM to display the muscles without their attachment points, and with a coarser cylinder representing their lines of action. SIMM performs this faster muscle drawing only when animating a motion continuously, using the **start** button in the Model Viewer. If you have many muscles in your model, this option can significantly increase the display of your model, making the animation playback much smoother. This option is on by default.

DISPLAY_ANIMATION_HZ

When this option is on, SIMM will display the frequency of the model display in the lower left corner of the model window, but only when animating a model using the **start** button in the Model Viewer. This frequency value, shown in frames per second, depends on the performance of your graphics card and the complexity of your model. This option is off by default.

DISPLAY POLYGONS PER SECOND

When this option is on, SIMM will display the number of polygons per second it is drawing, in the lower left corner of the model window, but only when animating a model using the **start** button in the Model Viewer. This value depends on the performance of your graphics card and the complexity of your model. This option is off by default.

DISPLAY_MOTION_INFO

This preference specifies whether or not to display information in the model window about the motion being animated. If it is on, SIMM will show the motion name, current time, and current frame in the lower right corner of the model window.

MOTION TRIAL IMPORT PREFERENCES

MOCAP MODEL

This preference specifies the model to use as the generic, unscaled model when importing a static trial to create a subject-specific model.

MARKER_HEEL_STRIKE_OFFSET

This preference specifies the offset from the local maximum of the marker trajectory to the time of the heel strike event. A heel strike event is defined as a transition from positive to negative in the distance between the sacral marker and the

heel marker (in the forward direction). For some subjects and marker placements it may be desirable to use a fixed offset from the local maximum rather than the maximum itself. The offset is specified in seconds, and affects the calculation of all marker-based heel strike events for all motion trials.

MARKER TOE OFF OFFSET

This preference specifies the offset from the local minimum of the marker trajectory to the time of the toe off event. A toe off event is defined as a transition from negative to positive in the distance between the sacral marker and the toe marker (in the forward direction). For some subjects and marker placements it may be desirable to use a fixed offset from the local minimum rather than the minimum itself. The offset is specified in seconds, and affects the calculation of all marker-based heel strike events for all motion trials.

MARKER EVENT TOLERANCE

This preference specifies the tolerance when determining if a marker-based event occurs at the same time as a force-based event. To calculate heel strike and toe off events for a motion trial, SIMM uses two methods: ground-reaction force vectors and marker trajectories. The force-based method is considered more accurate, but the marker-based method can differentiate between left and right foot events. When a marker-based event is within the specified tolerance of a force-based event, the events are considered to be identical. The resulting event uses the time of the force event, and the side (left/right) of the marker event. This tolerance can be relatively large (e.g., 0.1 seconds) to account for noise in the marker data without sacrificing the accuracy of the calculations.

EMG SMOOTHING WINDOW

This preference specifies the size of the moving window, in seconds, that is used to smooth raw EMG data. The value is used to initialize the appropriate field in the motion-import dialog box, but you can change it as you load individual motion trials

EMG SMOOTHING PASSES

This preference specifies the number of smoothing passes performed on raw EMG data. The value is used to initialize the appropriate field in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP INCREMENT

This preference specifies the frame increment to use when importing motion trials. A value of 1 means to import every frame (a value of 2 means to import every other frame). The value is used to initialize the appropriate field in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP SHOW MARKERS

This preference specifies whether or not to include the global XYZ marker positions in the motion created when importing a TRB/TRC/C3D file. If they are included, you can display them when animating the motion, in order to determine how well the model is tracking the markers. The value is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP CROP ENDS

This preference specifies whether or not to crop incomplete frames from the beginning and end of a motion trial when importing a TRB/TRC/C3D file. When this preference is *yes*, frames at the ends of the motion that are missing one or more markers are removed from the output. The value is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP CALC GAIT EVENTS

This preference specifies whether or not to calculate heel strike and toe off events when importing a TRB/TRC/C3D file. The value is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials. For information about how the events are calculated, see Section 5.2, Opening Tracked Marker Files.

MOCAP QUICK SOLVE

This preference specifies whether or not to use the Gauss-Newton solver when importing a TRB/TRC/C3D file. The Gauss-Newton method is usually faster than the default method (Levenberg-Marquardt), but can be less robust when some of the markers are missing. The value is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials. For most applications you can set this preference to *off*.

MOCAP CALC DERIVATIVES

This preference specifies whether or not to calculate derivatives of the imported data when loading a TRB/TRC/C3D file. If the preference is on, SIMM will calculate first derivatives of the gencoords, forces, and EMG data, and make them available for plotting in the Plot Maker. The value is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials. Unless you want to create plots of these derivatives, you should set this preference to *off*. If you use the motion as input to an inverse dynamic simulation (for which derivatives are needed), the simulation will calculate its own derivatives, so you do not need to calculate them when importing the motion.

MOCAP X AXIS

This preference specifies the units of the X axis when importing a TRB/TRC/C3D file. For most applications you should set it to TIME, so that event calculation and real-time playback give better results. But you can also set it to FRAME_NUMBER, allowing you to plot motion data as a function of frame number. This preference is used to initialize the appropriate field in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP START AT ZERO

This preference specifies whether or not the imported motion should start at time = 0.0 (or frame number = 1, if appropriate). If it is on, then after cropping frames or importing only a subset of them, SIMM will reset the time values in all frames so that the first frame has time = 0.0. If this preference is off,

then the imported frames will preserve whatever time values they originally had in the TRB/TRC/C3D file. This preference is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP LOAD ANALOG

This preference specifies whether or not to load analog data when importing a TRB/TRC/C3D file. When importing a TRB or TRC file, SIMM will look for an ANB or ANC file with the same base name. When a importing C3D file, SIMM will read the analog data directly from the C3D file. This preference is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP CALIBRATE FORCES

This preference specifies whether or not to calibrate the forceplate forces when importing a motion trial. To calibrate forces, SIMM determines the baseline of each forceplate channel and subtracts these values from the data, thus "zeroing" out the force data. For more information on force calibration, see Section 5.2, Opening Tracked Marker Files. This preference is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP MARKER NAME SOURCE

For C3D import only: This preference specifies whether the marker names for the motion trial should be read from the POINT:LABELS field or the POINT:DESCRIPTIONS field in the C3D file. This preference is used to initialize the appropriate field in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP SAVE MOTION FILE

This preference specifies whether or not to save a motion file when importing a TRB/TRC/C3D file. This file contains exactly the same data that is in the motion that SIMM loads onto the model when importing. This preference is used to initialize the appropriate checkbox in the motion-import dia-

log box, but you can change it as you load individual motion trials.

MOCAP MOTION FILE

This preference specifies the name of the motion file to create when importing a TRB/TRC/C3D file. For most applications you do not want to explicitly set this preference, because every time you import a motion trial the saved motion file will have the same name (unless you browse for a different name). Instead, do not define this preference, and let SIMM determine an appropriate motion file name for each motion trial that you import. This preference is useful primarily when creating scripts to batch process several motion trials.

MOCAP SAVE HTR FILE

This preference specifies whether or not to save an HTR file when importing a TRB/TRC/C3D file. HTR files cannot be read into SIMM, but they can be useful if you want to import the joint angles that SIMM calculates into another software package. This preference is used to initialize the appropriate checkbox in the motion-import dialog box, but you can change it as you load individual motion trials.

MOCAP HTR FILE

This preference specifies the name of the HTR file to create when importing a TRB/TRC/C3D file. For most applications you do not want to explicitly set this preference, because every time you import a motion trial the saved HTR file will have the same name (unless you browse for a different name). Instead, do not define this preference, and let SIMM determine an appropriate HTR file name for each motion trial that you import. This preference is useful primarily when creating scripts to batch process several motion trials.

MOCAP LIVE MODEL UPDATE

This preference specifies whether or not SIMM will update the model pose and refresh the display as it performs inverse kinematics on each frame of data in a motion trial.

MOCAP LOAD PERSONAL DAT

This preference specifies whether or not to load *personal.dat* with each motion trial. If this preference is set to yes, SIMM

will read *personal.dat* after importing the motion and update the subject information in the Motion Reporter.

MOCAP USE BAYES FILTER

This preference specifies whether or not to use a Bayes filter on EMG data when importing a motion trial. The Bayes-filtered data is stored in the output motion in an auxiliary data column that is linked to the EMG column (e.g., "semimem_r" and "semimem_r_aux"), and it is used to calculate onset time, offset time, and average amplitude for the EMG. If the Bayes filter is turned off, SIMM will use the moving-window smoothing algorithm from early versions of the program.

MOCAP ADAPTIVE MVCS

This preference controls how SIMM scales EMG signals. When this preference is on, SIMM will calculate the maximum value of each EMG channel for the motion trial. If the actual maximum is greater than the MVC specified in *personal.dat* (if any), SIMM will use the maximum as the MVC instead of the specified value.

MOCAP MIN FORCE

This preference defines the minimum allowable force when calculating force plate forces. Forces below this threshold are set to zero. If this preference is not specified, a value of 40.0 is used.

C3D MARKER RESIDUAL THRESHOLD

This preference defines the maximum allowable residual when importing XYZ marker data from C3D files. Marker coordinates with a residual above this threshold are treated as missing data. If this preference is not specified, a value of 10.0 is used.

FORCEPLATE VERTICAL OFFSET

This preference specifies a vertical offset that is applied to the display of the force plates in the model window when a motion trial is first imported into SIMM. A typical value is 0.001 (1 mm) so that the force plates are visible slightly above the floor. It does not affect the force calculations nor the display of the force vector in the model window. If this preference is not specified, a value of 0.001 is used.

STATIC TRIAL IMPORT PREFERENCES

MOCAP STATIC SUBJECT MASS

This preference specifies the mass of the subject, when importing a static trial to create a subject-specific model. The value is used to initialize the appropriate field in the static-import dialog box, but you can change it as you load individual static trials.

MOCAP STATIC LOAD PERSONAL DAT

This preference specifies whether or not to load *personal.dat* when importing a static trial. This file can contain information about subject parameters (*e.g.*, body mass, foot length, MVIC values) and model calculations (*e.g.*, hip and shoulder center positions). For more information on *personal.dat*, see Section 5.3, Using the Mocap Model. This preference is used to initialize the appropriate checkbox in the static-import dialog box, but you can change it as you load individual static trials.

MOCAP STATIC PRESERVE MASS DISTRIBUTION

This preference specifies, when importing a static trial, whether or not the mass distribution in the unscaled model should be preserved in the scaled model (the alternative is to scale the mass of each body segment based on its geometric scaling). For more information on scaling mass parameters, see Section 5.3.4, Scaling the Mocap Model. This preference is used to initialize the appropriate checkbox in the static-import dialog box, but you can change it as you load individual static trials

MOCAP STATIC MARKER NAME SOURCE

For C3D import only: This preference specifies whether the marker names for the static trial should be read from the POINT: LABELS field or the POINT: DESCRIPTIONS field in the C3D file. This preference is used to initialize the appropriate field in the static-import dialog box, but you can change it as you load individual motion trials.

MOCAP STATIC SAVE JOINT FILE

This preference specifies whether or not to save a joint file for the scaled model, when importing a static trial. This preference is used to initialize the appropriate checkbox in the static-import dialog box, but you can change it as you load individual static trials.

MOCAP_STATIC_JOINT_FILE

This preference specifies the name of the joint file to create when importing a static trial. For most applications you do not want to explicitly set this preference, because every time you import a static trial the saved joint file will have the same name (unless you browse for a different name). Instead, do not define this preference, and let SIMM determine an appropriate HTR file name for each motion trial that you import. This preference is useful primarily when creating scripts to batch process several motion trials.

MOCAP STATIC SAVE MUSCLE FILE

This preference specifies whether or not to save a muscle file for the scaled model, when importing a static trial. This preference is used to initialize the appropriate checkbox in the static-import dialog box, but you can change it as you load individual static trials

MOCAP STATIC MUSCLE FILE

This preference specifies the name of the muscle file to create when importing a static trial. For most applications you do not want to explicitly set this preference, because every time you import a static trial the saved muscle file will have the same name (unless you browse for a different name). Instead, do not define this preference, and let SIMM determine an appropriate HTR file name for each motion trial that you import. This preference is useful primarily when creating scripts to batch process several motion trials.

TOOL PREFERENCES

TIE_PM_OPTIONS_TO_PLOT

This option causes the Plot Maker to change its X-variable and Y-variable settings according to the current plot. If you have several plots on the screen which involve different X- and Y-variables, then when you use the plot selector to set the Plot Maker to a particular one, the X- and Y-variable settings

will change to the ones used in the plot you are choosing. Thus you can add additional curves (of the same type) to the plot without resetting these variables.

USE DEFAULT POSE FOR NORMALIZATION

This preference specifies which body pose to use when normalizing musculotendon lengths. In the Plot Maker, there is a Y-variable called *norm musculotendon length* that calculates the length of a muscle and then divides it by the length in the anatomical position (all gencoords = 0.0). If this preference is set to *yes*, then instead of the anatomical position SIMM will use the default pose (the pose that is applied when you select *restore gc values* in the Model Viewer).

SNAPSHOT FILE NAME

This option specifies the base file name to use when creating snapshot files of models in the Model Viewer. See the description of the **snapshot** command in Section 2.4.2, Command Menu, for more details. By default this option is set to "snapshot." To use a different file name, delete the default and type in a new name.

SCRIPTING CURRENT FOLDER

This preference specifies the current folder within the Scripting Tool. This is the folder that is used when opening and saving files within the tool, and can be changed within the tool using the *cd* command.

MOTION MOVIE SPEED

This preference affects how SIMM creates movies of motions, using the *movie* button in the Model Viewer. When MOTION_MOVIE_SPEED is defined as a positive number, SIMM will create one movie frame for each frame of data in the motion, regardless of the current *motion speed* setting and how long it takes to display one frame of data. After creating the movie, SIMM will modify the frame rate of the movie file so that it plays back at the speed specified by this preference. For example, if MOTION_MOVIE_SPEED is defined as 50.0, the resulting movie will play at 50% of the motion's original speed (as defined by the time column). If this preference is not specified or is not a positive number, the *motion speed*

setting in the Model Viewer is used to set the playback speed of the movie.

CONFIRM REPORT SIMULATIONS

This preference specifies whether or not to display a confirmation dialog box that must be dismissed before generating a motion report. If this preference is on, the dialog is displayed when *generate report* is selected if the report requires the running of at least one dynamic simulation to calculate kinetics variables.

NORM IMPORT TOLERANCE

This preference specifies the vertex tolerance when importing certain bone files into SIMM. When a bone file is read into SIMM, if it does not already contain a normal vector for each vertex then it is processed by Norm to calculate them. During processing, Norm merges all vertices that are within NORM_IMPORT_TOLERANCE of each other. If this preference is not specified, the default vertex tolerance is 0.0 (which means no merging will occur).

AUTO MOVIE FRAME RATE

This preference specifies the frame rate of movies generated using *auto movie* mode in the Model Viewer. Once *auto movie* mode is turned off and the movie file is saved, SIMM will modify the frame rate in the file to this value. If this preference is not specified, a value of 10.0 (frames/second) is used.

SKIN EXPORT TO MAYA

This preference is used when exporting skins as Collada files that will be imported into Maya. Due to an incompatibility between some versions of Maya and the open-source Collada parser, the units specified in the file need to be modified in order for the skin to load properly. If this preference is off, the following line will be written to the Collada file:

```
<unit name="meter" meter="0.001">
```

If this preference is on, this line will be written instead:

```
<unit name="meter" meter="1.0">
```

If you are creating the Collada skin file to import back into SIMM, this preference should be off (the default). If you intend to import the file into Maya, this preference should be on.

DISPLAY FORCES AS ARROWS

This preference specifies whether or not to display body forces, joint forces, and gencoord moments in the model window as arrows or as spheres. When this preference is on, each force and moment is displayed as a single arrow. When this preference is off, the X, Y, and Z components of each force and moment are display as concentric, translucent, red, green, and blue spheres. If this preference is not specified, forces and moments are displayed as arrows.

DISPLAY FORCES SPHERE SIZE

This preference specifies the size of the spheres when forces and moments are displayed as spheres (using the DISPLAY_FORCES_AS_ARROWS preference). The radius of the sphere displayed in the model window for a given force component is 0.05 times this preference times the magnitude of the component.

MOTION SMOOTHING FREQUENCY

This preference specifies the default low-pass cutoff frequency when smoothing columns of data in motions. This default value is used only when smoothing motion data with a script command that does not explicitly specify a cutoff frequency, or when the cutoff frequency is improperly specified in the motion smoothing dialog.

HIDE_DIALOGS

This preference specifies whether or not to suppress the display of most warning and confirmation dialog boxes in SIMM. Specifically, all dialogs that would be displayed while executing script commands will be suppressed, as this preference is meant to be used in conjunction with automated SIMM scripts.

NURBS SPRING SEPARATION

This preference is used when creating a spring contact object from a NURBS bone. It specifies the maximum allowable distance between adjacent springs when placing them on the surface of the object. If this preference is not specified, a default value is calculated using the length of the diagonal of the object's bounding box.

B.2 Colors

You can change any of the colors in SIMM by editing the color file, called *simmcolors.txt*. SIMM reads this file when the program is launched, and you can also reload it while SIMM is running by choosing Options... Reload Color File from the SIMM menu bar. When SIMM reads the file, it first looks for it in your user application data folder (e.g., C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\7.0\simmcolors.txt). If it is not found there it looks in the resources folder where SIMM was installed (e.g., C:\Program Files (x86)\MusculoGraphics\SIMM70\Resources\simmcolors.txt). To change a color in SIMM, open the file in a text editor and find the name of the item whose color you want to change. After the name are listed the red, green, and blue components, respectively, of the color. The numbers can range from 0.0 to 1.0. Change the color components as desired and then reload the color file

Note: model-specific colors and materials are specified in the joint file for that model, not in *simmcolors.txt*. See Section 3.3, Joint Files, and Section 3.3.7, Materials, for more information on changing model-specific colors and materials.

B.3 Password.txt

The first time you run SIMM on your computer you will be prompted to enter your SIMM password. At that point SIMM

will automatically store your password in a file named *password.txt*. This file is created by SIMM as a convenience so you will not need to reenter your password each time you run the program. SIMM attempts to create the file in the resources folder of your SIMM installation (*e.g.*, *C:\Program Files* (x86)\MusculoGraphics\SIMM70\Resources\password.txt), but if that folder is write-protected it will create it in your user application data folder (*e.g.*, *C:\Users\Peter\AppData\Local\MusculoGraphics\SIMM\7.0\password.txt*).

B.4 SystemInfo.txt

Each time SIMM runs it creates a file named *SystemInfo.txt* containing information about your computer. This information can be useful when trying to resolve technical problems with SIMM. If you request technical support by sending email to peter@musculographics.com, please include a copy of *SystemInfo.txt* with your message. The file is located in your user application data folder (e.g., C:\Users\Peter\App-Data\Local\MusculoGraphics\SIMM\7.0\SystemInfo.txt).

B.5 Bones

The *bones* subdirectory in the SIMM resources directory contains the bone files used to display the SIMM demo model. The bones in this directory are in the "newer" ASCII format. If you want to use them in another model that you are developing, you should copy them to a *bones* directory located in the directory containing your joint and muscle files (see Section 3.2, Opening Files, for more details).

\mathbf{A}	IGES format 230
act override (toggle button) 51 activation, muscle defined 63 specifying in file 282 active (toggle button) 51 analog data files opening 321 animations, see movies animations, see snapshots AVI files, see movies	opening 12 saving 19 SIMM ASCII format 224 SIMM binary format 227 SIMM old ASCII format 225 STL format 229 Wavefront OBJ format 228 bones deleting 140 editing 146 exporting 140 moving to different body segment 140
B body segment groups changing drawmode 24 defining in joint file 239	reading from files 142 selecting 145 specifying in body segments 238 writing to files 140
body segments 238–241 changing display parameters 133 restoring 128 saving to buffer 128 setting inertial parameters 129 Bone Editor 137–149 boolean operations 138 command menu 138 deleting polygons 140 deleting vertices 140 flipping normals 140 information header region 138 model-window commands 145 running norm on bone 138 splitting polygons 138 unselecting polygons 140 bone files 222–230	C C3D files see tracked marker files cameras restoring position of 22 saving position of 22 specifying in joints file 257 capturing images, see snapshots color factor using to display muscles 60 colors setting in simmcolors file 16 Constraint Editor 116–126 information header region 116 model-window commands 121
Collada format 230	objects command menu 117 points command menu 124
bone files 222–230	objects command menu 117

constraint objects	editing attributes 97
creating 117	inner and outer boxes 93
defined 116	specifying in joint file 263
deleting 118	tips and techniques 107
restoring 118	transforming 101
saving 118	deformities
specifying in joint file 270	specifying in joint file 267
transforming 123	deleting
constraint points	bones 140
creating 124	deleting markers 111
tolerance 126	drag and drop of input files 15
contact objects	drawmode
adding to body segments 130	flat shading 239
creating markers 110	gouraud_shading 239
cropping motions 166	none 239
cropping motions 100	of body segment groups 24
	outlined 239
D	setting in Segment Editor 128
_	setting in Segment Editor 128
def_bone 239, 255, 256	solid fill 239
default_value	-
of generalized coordinates 74, 247	specifying in joint file 238 wireframe 239
of materials 258	
Deform Editor 93–108	dynamic preferences 157
attributes panel 97	dynamics
command menu 96	input and output options 153
information header region 95	output variables 159
model-window commands 99	running in SIMM 149
tips and techniques 107	smoothing input data 156
transform panel 101	Dynamics Tool 149–162
deform objects	information header region 153
auto-reset deform objects 104	
combining multiple 95, 103	${f E}$
creating 97	£
defined 93	ellipsoid wrapping methods 90-92
deformity sliders 95, 106	axial 92
deleting 97	hybrid 91

midpoint 91 specifying in Wrap Editor 87 events deleting 165	clamping 30, 73 default value 74, 247 defining restraint functions 247 locking 30, 73 restoring from buffer 72
fiber length, defined 44 fixed segment changing interactively 24 fixed segment, defined 242 flat_shading, defined 239 force mattes adding to body segment 129 force, defined 44	saving to buffer 72 slider display 25 specifying velocities 293 tolerances for 74 typing in values 74 gouraud_shading, defined 239 gravity in dynamic simulations 157 specifying in joint file 236
form	TT
defined 9	H
editing 9	hot keys 35
forward dynamics 150	
Function Editor 189–195	I
functions 249–254	_
	IKS files 327
G	image capture, see snapshots
	inertia matrix
gait (toggle button) 180	specifying for body segments 129
gait events	inertial parameters
calculating 317	setting 129
Gencoord Editor 71–77	inverse dynamics 150
command menu 72	inverse kinematics
dof form 74	defined 76
information header region 72	specifying in joint file 234
generalized coordinate groups	turning on/off 76
defining in joint file 248	isometric (toggle button) 52
slider display 25	
generalized coordinates 245–248	

J	creating 110
Joint Editor 53–??	deleting 111
command menu 53	fixed 112
dof form 55	moving 115
information header region 53	name 111
rotation axes form 56	offset 112
transform order 56	radius 113
joint files 231–268	restoring from buffer 111
5	saving to buffer 110
opening 12	segment of attachment 112
saving 17	selecting 114
joints 243–245	unselecting 111
restoring from buffer 54	visible 112
saving to buffer 54	weight 112
	mass
L	specifying for body segments 129
	mass center
ligaments 286	specifying for body segments 129
linear function	material properties
defined 249	ambient 256
live model update 318	diffuse 256
live plot curves 81	emission 256
look at command 24	specular 256
	materials
M	assigning to body segments 238
M	assigning to bones 238
making movies, see movies	assigning to world objects 254
marker cloud	default 256, 258
defined 311	defined 256
Marker Editor 108–115	properties, see material properties
command menu 110	setting in Segment Editor 128
information header region 110	max positive moment, defined 45
model-window commands 113	maximum isometric force, defined 63
moving markers 115	menu, defined 7
selecting markers 114	mocap model 328-340
markers	choosing 330

joint center calculations 335	motion files 286–297
marker set 338	opening 13
scaling 336	saving 17
static pose 331	specifying in joint file 232, 287
model archive files 300	Motion Module 311–345
opening 14	importing IKS files 327
saving 19	opening analog data files 321
model display	opening C3D files 312
animating motions 30, 33	opening TRB/C files 312
moving joints 34	opening XLS files 321
muscle attachment points 31	real-time import 324
rotating 32	motion objects 293-296
trackball method 31	defined 287
zooming and panning 31	specifying in joint file 268
model poses	specifying in motion file 293
restoring 22	Motion Reporter 172–181
saving 23	command menu 176
specifying in joint file 275	output file form 179
model selector, defined 10	subject information form 179
model view, see camera	motions
Model Viewer 20–36	adding events 166
generalized coordinate form 28	animating 30, 33
information header region 20	cropping 166
model name form 29	deleting 165
model window viewing commands 31	displaying continuously 23
moving joints 34	dividing into steps 165
rotating the model 32	editing 165
sliders 30	normalizing 164
model-based script file	playback options 167
specifying in joint file 233	saving to files 164
moment arm, defined 43	smoothing 165
moment, defined 42	movie files 303
moment@maxforce, defined 44	movies 27–28
Motion Editor 163–172	auto loading 318
command menu 163	auto-movie mode 27
information header region 163, 176	choosing AVI file 27

motion movies 28	displaying in model window 23
using default file name 27	editing groups 60
Muscle Editor 57–71	including in dynamics 156
command menu 58	making groups 60
force parameters panel 61	moving attachment points 70
information header region 57	parameters, see muscle parameters
model-window commands 68	restoring from buffer 59
moving muscle attachment points 70	saving to buffer 59
selecting muscle attachment points 69	selecting attachment points 69
muscle files 277–286	surfaces 284
opening 12	musculotendon length, defined 42
saving 17	musculotendon velocity, defined 43
specifying in joint file 232, 277	
muscle force functions	* T
editing 61	${f N}$
muscle groups	natural cubic spline
in dynamic simulations 157	defined 250
muscle menu, defined 8	norm 305–310
muscle orientation, defined 45	options in Bone Editor 142
muscle parameters	norm fiber length, defined 44
activation, defined 63	norm musculotendon length, defined 42
editing 62	numerical moment arm, defined 43
maximum isometric force, defined 63	NURBS objects
maximum thickness, defined 283	adding to body segments 132
minimum material, defined 283	NURBS spring objects
minimum thickness, defined 283	adding to body segments 132
optimal fiber length, defined 63	5 5 5
pennation angle, defined 63	
tendon slack length, defined 63	0
muscle wrapping, see wrap objects	offsetting plot curves 50
	opening
muscles	bone files 12
adding attachment points 59	model archive files 14
deleting 60	motion files 13
deleting attachment points 60	muscle files 12
displaying attachment points 31	

plot files 14	toggle buttons 50, 180
script files 14	X-variables 46
tracked marker files 312	Y-variables 42
opening files 11	plot points
optimal fiber length, defined 63	creating 81
other data	plot selector, defined 10
in motion files 296	Plot Viewer 78–81
outlined drawmode, defined 239	plot window commands 80
output strides 181	plots
output strides (normalized) 181	adding labels 80
·	calculating means 78
_	creating, see Plot Maker
P	deleting 39
passive (toggle button) 51	deleting labels 78
password.txt file 16	deleting points 78
patches	resetting view 78
selecting 147	saving to files 80
pennation angle, defined 63	sending to model windows 80
plot curves	showing auxiliary data 79
clipping 49	showing cursor 78
creating, see Plot Maker	showing data points 79
deleting 39	showing events 79
making live 81	showing grid 79
naming 48	showing zero line 79
rectifying 51	polygons
plot files 298–300	selecting 147
opening 14	preferences
saving 18	in dynamic simulations 157
Plot Maker 37–52	preferences file 1
command menu 39	
generating curves from a model 40	0
information header region 37	Q
offset field 50	quick solve
options form 47	defined 316
plotting data from motion files 41	
scale field 50	

R	saving files 17
	scaling plot curves 50
real-time import see Motion Module	script files 301
	opening 14
rectify (toggle button) 51	saving 19
restoring	Scripting Tool 195–219
body segments 128	Segment Editor 126–137
camera positions 22	command menu 128
generalized coordinates 72	information header region 126
joints 54	model-window commands 134
markers 111	setting inertial parameters 129
model poses 22	segments
muscles 59 skins 183	editing 135
restraint functions 251	selecting 135
	selector menu 10
making active/inactive 73	shadows
	defining 239
S	displaying 31
	specifying color of 240
saving	Skin Editor 181–189
body segments 128	attachments panel 185
bone files 19	command menu 183
camera positions 22	group weighting 186
current muscle 59	information header region 182
generalized coordinates 72	model-window commands 187
joint files 17	unselecting skin points 184
joints 54	skin vertices
markers 110	changing attachment weights 185, 186
model archive files 19	group weighting 186
model poses 23	skins
motion files 17	editing attachments 185
muscle files 17	exporting as bones 184
muscles 59	making 184
plot files 18	restoring 183
script files 19	saving to buffer 183
skins 183	saving to files 184

unselecting points 184	TRB/TRC files
slider, defined 10	see tracked marker files
snapshots 25–27	
auto-snapshot mode 26	TT
choosing snapshot file 26	\mathbf{U}
including an alpha channel 26	units
motion snapshots 27	specifying in input files 222
taking 26	unselecting markers 111
solid_fill, defined 239	
spring floors	
adding to body segments 131	\mathbf{V}
spring points	velocities
adding to body segments 131	of muscle fibers 52, 279
moving 136	specifying in motion file 293
selecting 135	vertices
standard deviations	moving 149
in motion files 297	selecting 148
standard deviations (toggle button) 180	č
static pose	
see Motion Module	\mathbf{W}
step function	wireframe, defined 239
defined 249	world objects 254–256
sum (toggle button) 50 surface normals 305	Wrap Editor 82–93
	command menu 83
systemInfo.txt file 17	information header region 83
	model-window commands 89
T	tips & techniques 92
_	wrap objects
tendon slack length, defined 63	creating 83
tendon strain, defined 44, 45	defined 82
TIFF images, see snapshots	specifying in joint file 259
toggle button, defined 10 tracked marker files	transforming 88
cropping ends 316	wrapping methods, see ellipsoid wrapping
opening 312	methods
opening 312	